Probabilistic Approaches for Intelligent AUV Localisation

Francesco Maurelli

Submitted for the degree of Doctor of Philosophy

Heriot-Watt University

School of Engineering and Physical Sciences

Institute of Sensors, Signals and Systems

May 2014

Abstract

This thesis studies the problem of intelligent localisation for an autonomous underwater vehicle (AUV). After an introduction about robot localisation and specific issues in the underwater domain, the thesis will focus on passive techniques for AUV localisation, highlighting experimental results and comparison among different techniques. Then, it will develop *active* techniques, which require intelligent decisions about the steps to undertake in order for the AUV to localise itself. The undertaken methodology consisted in three stages: theoretical analysis of the problem, tests with a simulation environment, integration in the robot architecture and field trials. The conclusions highlight applications and scenarios where the developed techniques have been successfully used or can be potentially used to enhance the results given by current techniques. The main contribution of this thesis is in the proposal of an active localisation module, which is able to determine the best set of action to be executed, in order to maximise the localisation results, in terms of time and efficiency.

to the rainbow

Acknowledgements

I would like to express my gratitude to Prof. Yvan Petillot, my first supervisor, who introduced me to the world of underwater robotics and closely followed my adventure. He was always available when I needed his advice, he encouraged me during this journey, and with his expertise, understanding, and vivid engagement in all the lab activities, he was a constant reference during my work.

I would like to thank Prof. David Lane, my second supervisor, for his helpful advice. He was able not just to give a scientific support, but also to picture the long-term vision in the field, thus helping me to see my work as a brick towards that vision.

A special gratitude is for Prof. Pere Ridao, University of Girona, who hosted me during my research stay. It was certainly a very positive and productive period. It allowed me to put forward some important cooperation, which resulted in joint publications.

My work has been partially funded by the Marie Curie Research Training Network $FREE_{sub}NET$. The network was formed by 15 partners all over Europe. I would like to acknowledge its importance for both my personal and professional growth. Most of my research has been the result of joint work with other partners. The group of ESRs - Early Stage Researchers - is still in contact after the official end of the project. This is a measure of the success and of the importance of this kind of EU frameworks.

I received the advice and help of many colleagues from the University of Girona. For this reason I would like to thank Aggelos Mallios, Prof. Joaquim Salvi and Dr. David Ribas.

The work of Fitsum Reda, a student in the VIBOT Erasmus Mundus Master, was helpful for section 3.4 and, together with Mesfin Adame, for section 2.5.

Section 2.6 is the result of a joint work with Jan Sliwka, from ENSTA-Bretagne. He developed the Set Membership Approach, and we then compared with the Particle

Filter approach developed by the author.

Regarding structure inspection, the velocity control approach described in 4.3.2 was developed by George Karras and Charalampos Bechlioulis, from National Technical University Athens, and it was compared with other approaches developed by the author.

For the SAUC-E competition - section 7.1.1 - I would like to acknowledge the work of all the people in the OSL lab, who put countless hours in a team effort.

I would like to thank SeeByte Ltd. and in particular Dr. Jonathan Evans, for giving me the opportunity to contribute in the development of the Prototype Autonomous Inspection Vehicle (PAIV), as outlined in section 7.1.2.

I would like to thank all my colleagues and friends at the Ocean Systems Laboratory, including the many visiting students. The time spent together was very good and created a link that goes behind the work. In particular I would like to thank Nicolas Valeyrie, for the many discussions we had on both research and other interests, and Joel Cartwright, for his excellent engineering skills and kind availability to help whenever there was a need.

A special acknowledgement goes to my family for the support they always provided me. They always encouraged me in my work and they were always present in my heart both in the difficult and in the favourable situations, although sometimes far in physical distance.

ACADEMIC REGISTRY
Research Thesis Submission

Name*:*	Francesco Maurelli		
School/PGI:	Engineering and Physical Science / Institute of Sensors, Signals and Systems		
Version: *(i.e. First, Resubmission, Final)*	Final	Degree Sought (Award **and** Subject area)	Doctor of Philosophy Electrical, Electronic and Computer Engineering

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate*:*		Date:	

Submission

Submitted By *(name in capitals):*	FRANCESCO MAURELLI
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals):*			
Method of Submission *(Handed in to SSC; posted through internal/external mail):*			
*E-thesis Submitted (**mandatory for final theses)***			
Signature:		Date:	

Please note this form should bound into the submitted thesis.

Updated February 2008, November 2008, February 2009, January 2011

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Day after day, the number of operations at sea has significantly increased in the last few years. This is related to many different fields, such as exploration, exploitation of resources (e.g. oil underwater infrastructures), security (e.g. harbour protection) and life sciences studies. Remotely Operated Vehicles (ROVs) are safely and routinely used in the off-shore industry, for underwater operations. Although they represent essential equipment for many tasks, limitations are high. They have no intelligence as they are driven by a human pilot, connected through an umbilical cable. This means a complete lack of autonomy and the need of costly infrastructures to operate with those vehicles. Human pilot is needed for all the time the ROV is operating in the field, and a support vessel is essential to deploy the vehicle. Additionally, due to the cable constraints, the support vessel needs to keep its position during the full operational time of the ROV. The operations are also limited by the connected cable. For example underwater caves or under-ice locations remain inaccessible. Autonomous Underwater Vehicles (AUVs) address these limitations as they do not need a human pilot and the costly infrastructures required by the ROVs. They are able to perform more complex missions, due to the absence of a connecting cable between the vehicle and the mother ship and the on-board capability of data processing and decision making. The development of the Intervention-AUVs (I-AUVs) concept led to vehicles that can also interact with the structures and be used for autonomous inspection and intervention.

A key area for intelligent vehicles is to correctly estimate the state of unmanned systems in the operational environment, i.e. the capability of the robot to correctly estimate its position and orientation. The problem is usually known as *autonomous localisation*.

Localisation techniques are required for many underwater applications involving autonomous and semi-autonomous robots. For both survey-class and intervention-class AUVs, a localisation system is a prerequisite for almost every task. Two very im-

portant topics are for example autonomous docking and navigation around/inspection of underwater structures. For docking purposes, it is important to correctly estimate the relative state of the vehicle with respect to the docking station. For inspection purposes, understanding its own location allows the robot to correctly navigate around underwater structures, both for inspection or intervention.

Being underwater intelligent vehicles, some of the research fields are similar to the ones for ground or aerial intelligent vehicles. The peculiar features of the environment are however different. Therefore different sensors and different strategies, according to the constraints of the underwater world, need to be designed and used.

The aim of this thesis is therefore to analyse current work in the intelligent localisation area, and develop novel approaches to overcome some of the shortfalls identified.

1.1 Thesis objectives and contributions

The research carried out as part of this thesis addresses the following areas:

1. investigate current localisation techniques for autonomous underwater vehicles;

2. design and realise localisation systems for AUV, addressing some of the shortcomings in current techniques, in terms of computational efficiency and precision;

3. link the localisation system of an AUV to the overall planning system, in order to get better localisation results.

Regarding 1, this thesis presents an up-to-date analysis of the state of the art, mainly in Chapter 2 and Chapter 5. Some techniques are also explained in details and implemented.

Regarding 2, the main contributions of this thesis are:

1. design, implementation and integration of a novel approach for underwater vehicles, based on improved Particle Filters, with a comparison with state-of-the-art techniques;

2. design and implementation of a mixed localisation system, using both Extended Kalman Filters and Particle Filters (3.4);

3. design, implementation and integration of novel approaches in localisation with respect to a structure (4).

Regarding 3, the main contributions of this thesis are:

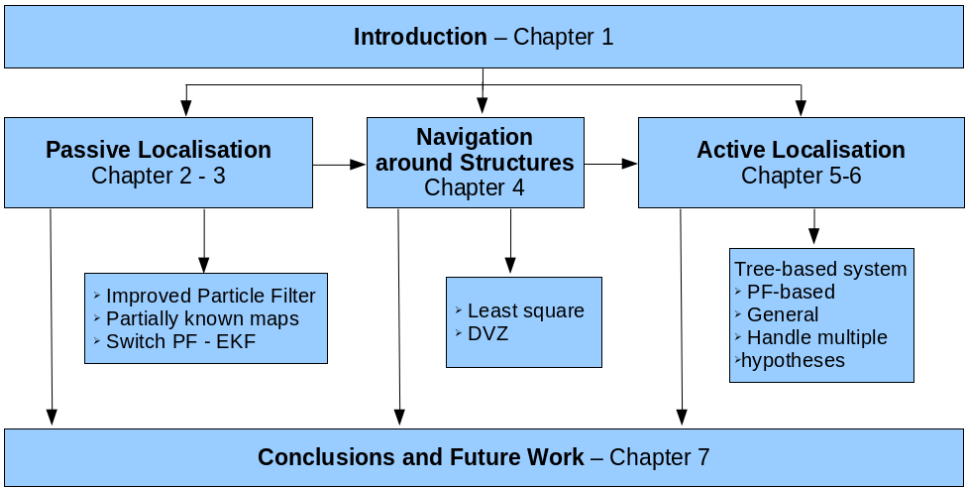


Figure 1.1: Thesis structure and relations among the Chapters.

1. design, implementation and integration of a novel framework for active localisation, using a tree-based planner, to aid localisation (6.2);

2. extension of the framework to be more adaptable and possible to be integrated in different systems (6.2).

Among the contributions presented in this thesis, the author considers the latter block as the most important, as related to the autonomy of the underwater vehicle and its decision-making capability, linked to the navigation and localisation capability. It is also important to underline that all proposed algorithms have been implemented and tested with real data, and in many cases fully integrated in the AUV architecture and tested in field trials.

Figure 1.1 presents the organisation of the thesis. The first block is a presentation of passive localisation techniques (Chapter 2), followed by a contribution in the field (Chapter 3). The need of incorporating the control of the vehicle in the navigation system has lead to the investigation of navigation around structures (Chapter 4). Considering that this solution was purely reactive, an analysis of deliberative active localisation systems is presented in Chapter 5, leading to a contribution part in Chapter 6, where a novel framework for active localisation is presented.

The Introduction (current Chapter) develops then with a brief presentation of the localisation problem, in terms of a probability formulation. Underwater sensors are presented, with special attention to the sonars used in the thesis. The vehicles employed and the different test sites and facilities are finally described.

1.2 Localisation

The problem of self localising a robot consists in determining its pose in the operating environment, given the observation history, the command history, and the knowledge of the environment, as presented in [107]. Analytically, it can be expressed as estimating the probability distribution:

$$p(x_t|z_{0:t}, u_{0:t}, m) \tag{1.1}$$

where x_t is the robot position in the environment at time t, $z_{0:t}$ is the sensing history $z_0, z_1, ..., z_t$ up to time t, $u_{0:t}$ is the command history $u_0, u_1, ..., u_t$ up to time t, and m is the map of the environment. The observation history $z_0, z_1, ..., z_t$ can be also referred to as exteroceptive sensing history, since it is relative to the exteroceptive sensors. Such devices are able to acquire external (with respect to the robot) measures. Two different approaches to localisation are analysed: position tracking and global localisation. In the first one, the new position is computed, assuming to have a good estimate of the previous one. Analytically, position tracking consists in estimating the following distribution

$$p(x_t|z_{t-1}, u_{t-1}, x_{t-1}, m) \tag{1.2}$$

while the global localisation can be expressed in terms of 1.1. Usually in position tracking the posterior about robot poses is often represented using a Gaussian distribution. This enables those approaches tracking only a single mode of the evaluated posterior. If the wrong mode is chosen, the estimate is extremely unlikely to converge to the correct solution. In contrast, global localisation approaches are able to deal with ambiguous situations, by representing the posterior about robot poses through a multi-modal distribution. This results in a greater robustness, but in an increased computational complexity.

Without any external aid or without sensing the environment, the only possibility for the robot is to use *dead reckoning*, which deals with state sensors only. They can provide relative information (e.g. speed, acceleration, etc.) or absolute information (e.g. heading, global position, etc.). The problem with these techniques is the growing error over time of the estimated position with relative sensors, like Doppler Velocity Log (DVL) and Inertial Navigation Systems (INS), while a global fix in position, through GPS is possible only on surface. The only possibility to improve the navigation using only those kind of sensors is either to improve the sensor quality or working operationally very close to surface.

An *et al.* demonstrated the fusion of differential GPS and INS at two feet below

water surface [4]. Their work focused successfully on the fusion between the two sensors and on noise removal.

Erol *et al.* proposed to use a GPS-aided localisation [28]. Considering that GPS signal does not propagate in the water, the AUV is therefore forced to acquire the signal at the surface. This approach is not very reliable since the vehicle has no access to GPS signal during the submerged period and has to estimate its state with other sensors. To handle this problem, the vehicle dives to a fixed depth and follows a predefined trajectory, which is not suitable for navigation in complex environments, and prevents the possibility of any in-mission real-time replanning.

Other navigation techniques need to be explored in order to perform operations without the constraint of being close to surface and, at the same time, correcting the error of techniques based on pure dead reckoning.

A commercially available solution is represented by acoustic aid. Acoustic techniques are widely used to aid localisation. The three main methods of calculating a position using acoustic are:

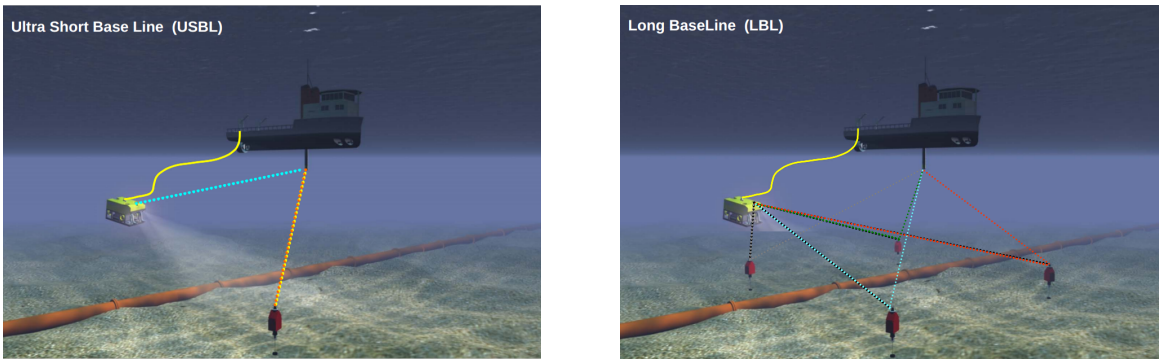
- **Long Base-Line (LBL)**
  Determines beacon position by measuring the slant ranges from three or more widely spaced transponders;

- **Short Base-Line (SBL)**
  Determines beacon position by measuring the relative arrival times at three or more vessel mounted hydrophones;

- **Ultra Short Base-Line (USBL)**
  Determines beacon position by measuring the relative phases of the acoustic signal received by closely spaced elements in a single hydrophone.

As outlined by Thomson in [106], the most precise of the three is the LBL system. Both SBL and USBL requires a one-time calibration at installation. On the other hand, LBL requires a calibration at each deployment. This is not however a massive drawback, as transponders for LBL can be deployed and remain active for long time without the need of doing anything. USBL requires only one transducer so the system complexity is lower compared to the other techniques. SBL requires multiple hydrophones, but this leads to multiple solutions and improved accuracy over USBL. In operational activities, LBL is generally used for the global localisation of the vehicle, and the transponders delimit the mission area. SBL/USBL is generally used for communication with a surface vessel. Figure 1.2 shows how USBL and LBL work for subsea operation.

However it is not always possible to deploy transponders and, even when it is possible, for navigation close to subsea structures, appropriate filtering techniques need



(a) USBL in action

(b) LBL in action

Figure 1.2: USBL and LBL systems: the first one has a direct contact with the ship, whilst the second one calculates its position using the deployed transponders.

to be used, as well as integration with sensor data. In the next section a brief introduction on the challenges of underwater sensing and some details on the underwater sensors employed are therefore presented.

1.3 Underwater Sensing

This section briefly presents the sensors used to acquire information about the environment.

1.3.1 Vision

In the robotic domain the vision system is generally considered one of the main sensors to acquire information about the world surrounding the robot. However, in the field of underwater robotics, vision systems have many limitations. Such systems work only in clear water. Water is about 800 times denser than air and this has impact in optical sensors and in the penetration rate. Additionally, scattering and refraction change the direction and the speed of light waves. The zone or depth at which light penetrates in water allowing plants to exist is known as the *Photic Zone*. The amount of light that penetrates the water depends on the amount of dissolved minerals, silt and detritus material contained in it (scattering and absorption) [73]. During the penetration, however, a portion of light is absorbed and converted into heat or used for photosynthesis. As light penetrates water, the colours are absorbed at different depths, as shown in Table 1.1.

Irregular sea surfaces affect visibility in several ways. Variable refraction results in a reduction of the contrast of a target. All these factors reduce camera visibility to a range of tens of meters, in the best conditions. Figure 1.3 illustrates light propagation

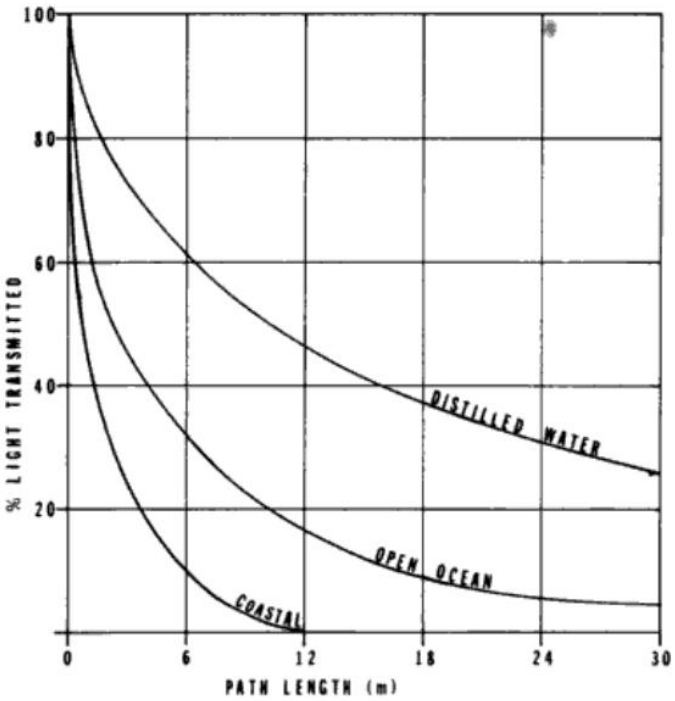


Figure 1.3: Light transmission and propagation according to different path length and water types.

underwater in different water types [27].

1.3.2 Laser

Laser scanners or LIDARs are also very often used in land robotics. However, their application for underwater scenarios is still limited. While there are a few approaches, like for example [89], [113], [22], underwater laser scanners have a very limited range and require very clear water for the light to propagate.

1.3.3 Sonar

Considering the limitations of usual sensors used in other robotic fields, in the underwater domain there is the need of an instrument that is able to address the

Colour	**Wavelength**	**Depth**
Red	780 to 622 nm	5m
Orange	622 to 597 nm	15m
Yellow	597 to 577 nm	30m
Green	577 to 492 nm	60m
Blue	492 to 455 nm	75m

Table 1.1: Wavelenghts and depth penetration of different lights.

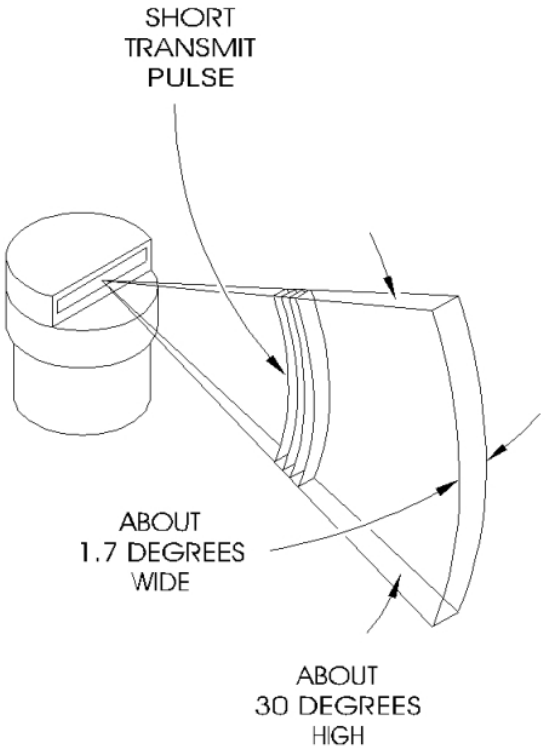


Figure 1.4: Typical fan shaped beam as used in imaging sonars [1].

peculiarities of the water medium. Sonar is a system that uses transmitted and reflected underwater sound waves to detect and locate submerged objects or measure the distances underwater. It stands for **SO**und **N**avigation **A**nd **R**anging. The first rudimentary sonar system has been described by Leonardo Da Vinci, in 1490: it was a simple tube used to detect vessels by placing an ear to the tube [29]. However the use of a sonar system to locate underwater obstacles comes only in the second decade of the 20^{th} century, with the world first patent for an underwater echo ranging device filed in 1912, a month after the sinking of the *Titanic*. Due to the conditions typical of the underwater environment, SONAR remains the main sensor used to acquire information about the environment.

Not being based on vision, a sonar system can have a range of several hundreds of meters. There are two main classes of sonar: profiling and imaging. In Imaging sonars, a fan-shaped sonar beam scans a given area, by either rotating or moving in a straight line, through a series of small steps, as explained in Figure 1.4. The beam's movement through the water generates points that form a sonar image of the given area.

In Profiling sonars, a narrow pencil-shaped sonar beam scans across the surface of a given area generating a single profile line on the display monitor, like explained in Figure 1.5. This line, consisting of a few thousand points, accurately describes the cross-section of the targeted area. A key to the Profiling process is the selection of

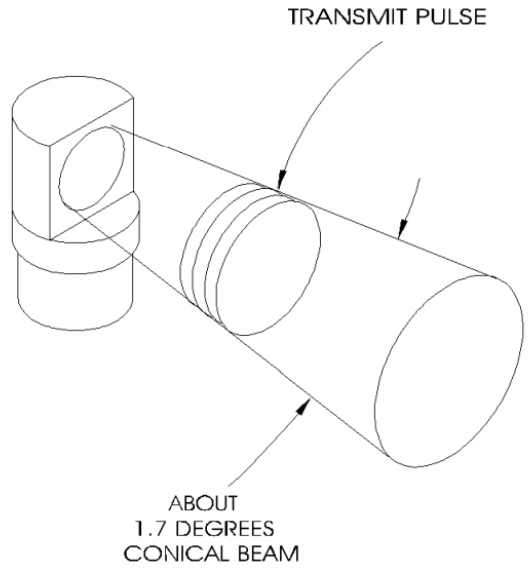


Figure 1.5: Typical pencil shaped beam as used in profiling sonars [1].

the echo returns for plotting. The sonar selects the echo returns, typically one or two returns for each "shot" based on a given criterion for the echo return strength and the minimum profiling range. The information gathered from the selection criteria forms a data set containing the range and bearing figures.

For the development of this thesis, imaging sonars have been employed, although they have often been treated as profilers, through appropriate signal and image processing techniques.

Table 1.2 presents the different sonars used in this thesis with their features.

Sonar **Range**	**Frequency** **Scanning/Multibeam**	**Field of view** **Vehicle**
Tritech SeaKing 300 m	325-650 kHz scanning	0-360deg PAIV AUV, Cartesian Robot
Tritech MiniKing 100 m	675 kHz scanning	0-360deg Ictineu AUV
Tritech Micron 75 m	700 kHz scanning	0-360deg Nessie IV AUV
Tritech Micron DST 300 m	325-650 kHz scanning	0-360deg Nessie V AUV
Tritech Gemini 720i 120 m	720 kHz multibeam	120deg Nessie V AUV

Table 1.2: The different sonar employed and their features.

1.4 Navigation Sensors

This section will present the sensors that a vehicle can use to understand its own motion.

1.4.1 Doppler Velocity Log

A Doppler Velocity Log (DVL) is a hydroacoustic sensor, which is useful for bottom-tracking. Using the Doppler effect of sound waves reflected back from the bottom, it is able to give an estimate of the velocity of the vehicle. In case the vehicle is at an altitude where it is impossible to have bottom tracking, the same principle of the DVL can be applied to try to estimate the movement of the vehicle in the water column, using the sound waves scattered back from particles within the water column. The position tracking of the vehicle is performed integrating over time the velocity. This naturally leads to an increased error in the position estimate.

1.4.2 Inertial Measurement Unit

An inertial measurement unit, or IMU, is an electronic device that measures and reports the vehicle's acceleration in different axes. According to the type and the cost involved, it can have a combination of accelerometers, gyroscopes and magnetometers. From the acceleration data, in order to get the position, a double integration over time is needed. Again, as for the DVL described in Section 1.4.1, the error is increasing over time.

1.4.3 Compass

A compass is a sensor which provides the orientation of the vehicle with respect to the magnetic north. Unlike the DVL and IMU, its error does not increase over time, as each measure is independent and provide a value in global world coordinate. The drawback however is that a compass can be easily influenced when navigating around metallic structures. The error of a compass for AUV can arrive easily to a few degrees, which can create serious problems for long transit.

1.4.4 Fibre Optic Gyroscope

A fibre optic gyroscope is a sensor that measures the orientation in one or multiple axes, considering the sensed acceleration. It is very precise and its value needs to be compensated for the Earth's rotation. Unlike the compass, it cannot provide any global value. It is therefore usually used in combination with a compass. The compass

is used to get the first initial global value, and then it is updated consequently to the gyroscope's readings, therefore providing orientation values much more accurate than using the compass only.

1.5 Underwater Vehicles

Unmanned underwater vehicles (UUV) are any vehicles that are able to operate underwater without a human occupant. These vehicles may be divided into two categories, Remotely operated underwater vehicles (ROVs), which are controlled by a remote human operator, and Autonomous underwater vehicles (AUVs), which operate independently of direct human input.

1.5.1 Remotely Operated Vehicles

Remotely Operated Vehicles - ROV - are human-piloted underwater vehicles and routinely daily used in the oil&gas industry. The main features are:

- the vehicle is not autonomous, neither has any cognition;
- the vehicle is connected through an *umbilical* cable, to receive power and for data exchange, including control commands;
- the commands sent by the human pilot are low-level. No high-level tasks are given to the vehicle.

External power allows use of powerful lights. Additionally, because of the possibility of data-transfer given by the cable, the pilot can receive lots of data real-time. The use of ROVs is however very expensive, as they require a support vessel and a human pilot for the full mission time. Other limitations come from the nature of the cable itself, making the exploration of complex structures very difficult, if not impossible. Under-ice missions are another example where ROVs are unsuitable.

1.5.2 Autonomous Underwater Vehicles

Autonomous Underwater Vehicles - AUVs - aim to address the shortcomings in the use of the ROVs, moving control from the pilot to the vehicle itself. An increased interest both from industry and from academia goes towards more reliable and more intelligent vehicles, aiming both at reducing operational cost and at increasing the range of different missions that can be successfully performed underwater. Without the *umbilical* cable, the vehicles need to have energy storage and computation capabilities. A consequent change deals with the shape of the robot: whilst ROVs are

often *ad hoc* used for industrial manipulation tasks, and therefore not particularly hydrodynamic, AUVs are generally used for long explorations, thus often designed to be torpedo-shaped.

Applications range from biological survey, to oceanography, to military applications. The robots became increasingly more reliable and equipped with better payload.

1.5.3 Intervention Autonomous Underwater Vehicles

The increased progress in autonomous underwater vehicles leads to the creation of a new field of vehicles, the so-called Intervention/Inspection-Autonomous Underwater Vehicles - I-AUVs. I-AUVs - sometimes called *Intelligent ROVs* aims to perform tasks currently performed by ROVs only, like close inspection of underwater structure and intervention. This thesis focuses both on AUVs and I-AUVs, positioning itself on the problem on underwater localisation for autonomous inspection.

1.6 Platforms

In this section, all the robotic platforms used for the development of this thesis will be briefly described.

1.6.1 Cartesian Robot

The Cartesian Robot (Figure 1.6) is an important asset to gather sensor data at different and controlled distances. It is actuated with a DMC-1380 motion controller, actuating three degrees of freedom, alongside the three geometrical axes of the pool. Sensors can be easily plugged in, for data gathering in controlled conditions. It is fully integrated with the vehicles' software architecture, based on ROS. The expected precision is about 1 cm. The operational capability spans out over the full dimension of the OSL tank, being it 3x4m and 2m depth. A pan & tilt unit can be mounted on the end effector increasing therefore the degrees of freedom of the sensors attached. Details on the use of this vehicle are in Chapter 3.

1.6.2 Nessie IV AUV

The vehicle is made up of two 22cm diameter cylindrical aluminium hulls surrounded by a Delrin polymer frame. This cage serves as a mounting point for sensors, protects the contained devices from impact, and keeps the thrusters safely out of the way of human divers. One hull, dubbed the motor hull, houses batteries and H-bridge

Figure 1.6: The Cartesian Robot in the OSL pool. Equipped with a DMC-1380 motor controller, it is actuated on three degrees of freedom. Sensors can be easily plugged in, for data gathering in controlled conditions.

controllers to drive the thrusters. The other, the PC hull, contains the embedded computers, interfacing and sensing electronics and the batteries to power these. Separating the power supplies and electronics in this way provides a degree of isolation from noise and power fluctuation caused by the H-bridges. It also ensures that even if the thrusters drain their batteries to a low level, the computers remain operational and control is still possible. A photograph of the submerged vehicle can be seen in Figure 1.7. The computers used in the vehicle are industrial MSM800 PC104 embedded PCs. It has a AMD Geode LX800 500 MHz Processor, 512 mb of RAM, 4 USB 2.0 ports, 2 serial ports and one 100 Mbit Ethernet port. Two of these embedded PCs are used in the vehicle; one for low level sensing and control, and another for video capture and processing. These are connected via Ethernet. This split ensures that that primary control PC is never starved of resources by the image processing algorithms. To make the computer more robust, flash based solid-state hard disks are used instead of standard magnetic disks. These further reduce the power requirements of the system and make it more robust to bumps and jerks as are expected in a mobile vehicle. Each of the vehicle's hulls contains two 10 cell Nickel- Metal-Hydride (NiMh) battery packs connected in series to provide a nominal 24 volts. The Tritech PA500 is used to measure the distance between the vehicle and the floor of the tank or the seabed. This has a maximum operating range of 50 metres. A Keller Series 33X depth sensor is used to measure the distance between the vehicle and the water surface. For a flat environment, this sensor gives an information very similar to the altimeter, but its precision close to the bottom is higher. A TCM 3 compass measures the vehicle's heading with a precision of 0.5 degrees. It is able to compensate for the vehicle's tilt up to 80 degrees.

The vehicle is fitted with four 50 metre rated underwater colour cameras. They interface to the second PC104 with a PC104 form-factor video capture board. This can capture up to 25 frames per second, at 24 bit colour depth. Two cameras are arranged in pairs, looking forward, for stereoscopic vision. The downward facing camera can be used for ground target detection, bottom mapping, feature extraction, while the upward facing camera can be used for ship hull inspection, cave or and under-ice inspections. An extremely compact Micron DST Sonar is mounted on the vehicle for obstacle avoidance and mapping. Vertical beamwidth is 35 deg and horizontal beamwidth is 3 deg. Details on the use of this vehicle are in Chapter 3, 4 6.

1.6.3 Nessie V AUV

Nessie V has been developed as a multi-functional open platform capable of accommodating different sets of sensors, and to perform a large variety of missions, such as the aforementioned types.

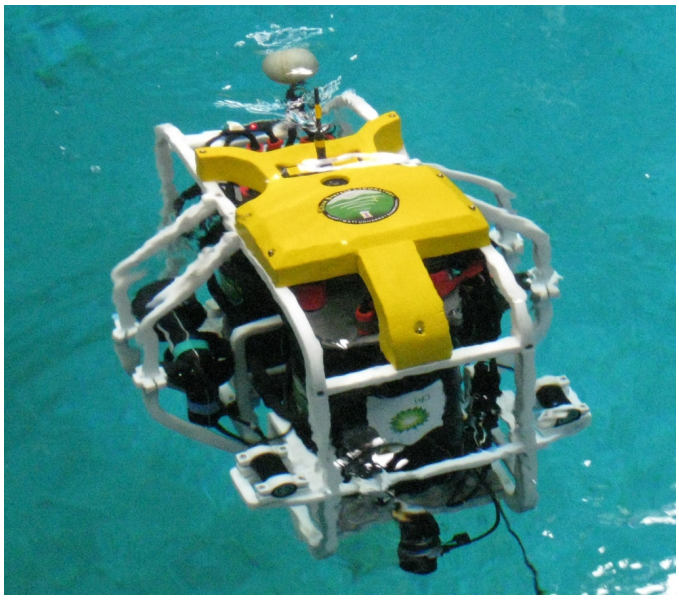


Figure 1.7: Nessie IV AUV. Fully actuated on four degrees of freedom, it mounts a Tritech Micron sonar, used for localisation.

In terms of size, shape and manoeuvrability, Nessie V is a compact autonomous underwater vehicle that combines a torpedo shape for fast transit with hover capability through the use of six SeaBotix thrusters. The maximum diameter is 30 cm long and the length is 175 cm. A range of missions from wide area surveys to the close inspection of objects of interest can be easily performed. Nessie V is equipped with a 2.25 kW power source comprised of four Lithium polymer battery packs, which provides approximately 12 hours of endurance when performing missions such as a pipe or a wall inspection. Station keeping in low tidal current areas can be maintained for several days. The sensors currently on-board include a forward looking Tritech Gemini sonar, a rotating Tritech Micron sonar, four colour underwater cameras, two omnidirectional hydrophones, a PNI TCM 6 compass, a KVH DSP-3000 fibre optic gyroscope, a Keller Series 33X depth sensor, a GPS, an acoustic WHOI Micromodem, and a Teledyne Explorer PA DVL. Nessie V is equipped with two industrial PC104 PCs chosen for their small size and low power consumption. They each have a 1.66 GHz Atom Processor, 1 GB of RAM, four USB 2.0 ports, four serial ports, and one gigabit Ethernet port. One PC has an on-board GPS receiver. The two PCs are connected via Ethernet, one being used for sensing and control, and the other for video capture and processing. Flash based solid-state hard disks are used in lieu of standard magnetic hard disks, for robustness. Details on the use of this vehicle are in Chapter 4 and 6.

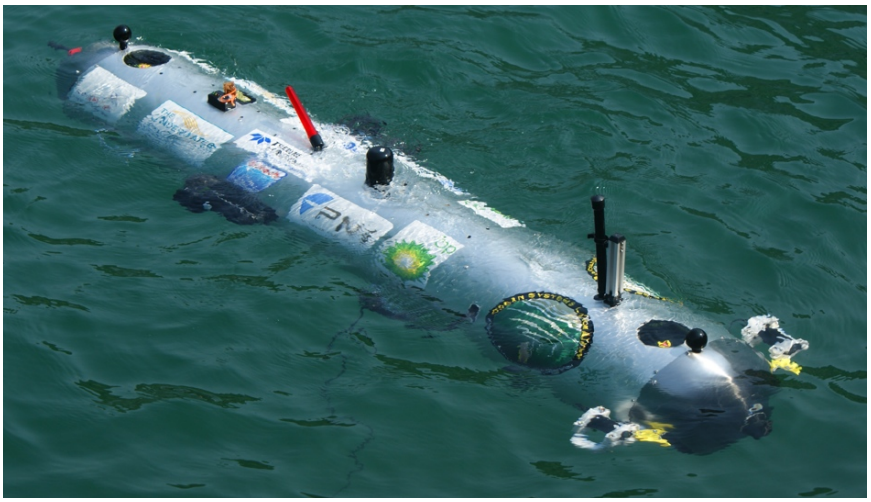


Figure 1.8: Nessie V AUV. Fully actuated on five degrees of freedom, it mounts a Tritech Gemini sonar, used for localisation.

1.6.4 Ictineu AUV

Ictineu AUV, in Figure 1.9 was developed by the University of Girona. Tailored around a typical open frame design, the vehicle is composed by two cylindrical pressure vessels made of aluminium, to host power and computer modules while a smaller one made of Delrin contains a Motion Reference Unit (MRU). It has four fully actuated degrees of freedom. The robot chassis is made of Delrin, an engineering plastic material which is lightweight, durable and resistant to liquids. Among the mounted sensors, the robot is equipped with a SonTek Argonaut Doppler Velocity Log (DVL) specially designed for applications which measure ocean currents, vehicle speed over ground and as an altimeter using its 3 acoustic beams. A pressure sensor provides water column pressure measurements and a Xsens MTi low cost miniature Attitude and Heading Reference System (AHRS) provides a 3D orientation (attitude and heading), 3D rate of turn as well as 3D acceleration measurements. The robot is equipped both with vision sensors (two cameras) and an acoustic sensor (Tritech Miniking Mechanically Scanned Imaging Sonar - MSIS), designed for use in underwater applications such as obstacle avoidance and target recognition. Dataset from a marina were gathered by Ictineu and used to test localisation algorithms described in Chapter 3. Those data export readings from the Inertial Measurement Unit (IMU), Doppler Velocity Log (DVL), and Tritech Miniking Imaging Sonar.

1.6.5 PAIV AUV

The Prototype Autonomous Inspection Vehicle (PAIV) is an underwater vehicle resulted from the joint efforts of SeeByte Ltd. - a spin-off company of the Ocean

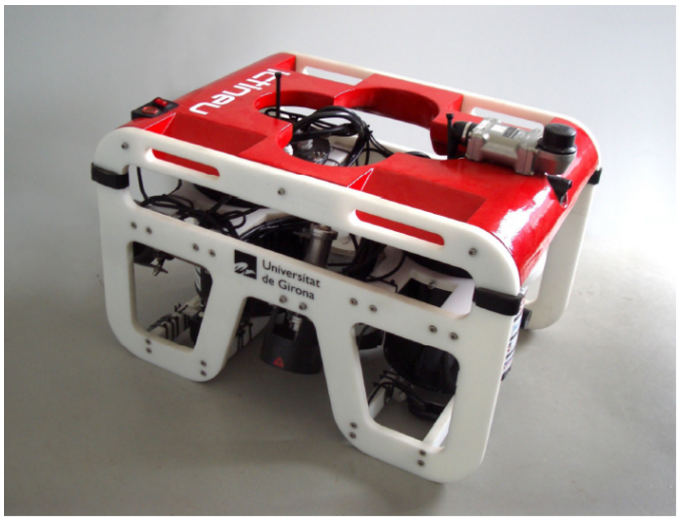


Figure 1.9: Ictineu AUV. Fully actuated on four degrees of freedom, it mounts a Tritech Miniking sonar, used for localisation.

Systems Lab. - and Subsea7, a large subsea engineering and construction company. The vehicle is rated 1,000 m depth (with easy upgrade up to 2,000), and it represents a prototype autonomous inspection vehicle. This vehicle represents the commercial industrial interest for a new kind of skilled AUV, who are capable of carrying out complex tasks, and not just preplanned surveys. With a weight of 350 Kg, and equipped with the best available sensors, it is considered in the industrial world as an innovative, light intelligent vehicle. It mounts a fibre optic gyroscope, depth and altimeter sensors, two cameras, a Tritech SeaKing sonar and can host a BlueView sonar according to the operational needs. Details on the use of this vehicle are in Chapter 7.

1.7 Test sites

In this section, all the test sites used for the development of this thesis will be briefly described.

1.7.1 OSL Tank

The OSL tank is a 3x4 m tank, 2 m deep. Located in the Ocean Systems Laboratory, it represents a comfortable and easy-to-use facility for fast in-water debugging and validation. Equipped with a Cartesian robot, described in 1.6.1 and with a manual crane, it is usually the first site used to test algorithms as well as basic hardware and software checks for new vehicles built in the laboratory. It has been the first test site for the algorithms described in this thesis, in Chapter 2, 3 and 6.

Figure 1.10: PAIV AUV. Fully actuated on four degrees of freedom, it mounts a Tritech SeaKing sonar, used for localisation.

Figure 1.11: OSL Tank. Used for several localisation tests, both with the Cartesian robot and with Nessie AUV

1.7.2 HWU Wave Tank

The HWU wave tank is a 10x12 m tank, 2.5 m deep. Located in the School of Built Environment of Heriot-Watt University, it provides a reasonably sized environment for more complex testing, which would be impossible to be carried out in the OSL small tank. It has a suspended *beach* for wave absorption on the long size, which is half a meter deep. This reduces the surface dimensions of the tank to 8.5x12 m. Equipped with an automatic crane, and with a bridge which allows easy accessibility to the centre of the tank, it represents a very valuable *in site* test site, even for complex tasks and for big vehicles. Additionally it is possible to start waves with different intensity and frequency, a feature that has been often exploited to test the vehicles' ability to overcome the disturbances given by currents and waves. It has been used to test the algorithms described in this thesis, both in Chapter 3, 4 and 6.

1.7.3 Subsea7 Test Tank

The Subsea7 Test Tank is a circular tank, with a diameter of 10 m, 8 m deep. It is located in Aberdeen, at Subsea7 headquarters. It has been used for navigation around structure and autonomous localisation, as presented in Section 7.1.2. Surface and underwater cameras are linked to a control room, where it is possible to comfortably operate.

Figure 1.12: HWU Wave Tank. Used to tests several localisation algorithms, it can produce waves at different frequences and amplitudes.

1.7.4 Somerton Diving Pool

The *JEM Divers* Pool, located in Somerton, is a 12x7 m, with a gradient descent from 1 to 2 m deep, and a final part 5 m deep. It has been the testing facility in preparation for the Student Autonomous Underwater Challenge - Europe (SAUC-E) 2009. The localisation algorithm described in Chapter 3 has been tested there, as outlined in Section 7.1.1.

1.7.5 QinetiQ Ocean Basin Tank

The QinetiQ Ocean Basin Tank is the the biggest covered water space in Europe, used for ship testing. It is 122m long, 61m wide and 5.5m deep, with a portion equipped for wave production. It hosted the Student Autonomous Underwater Challenge - Europe (SAUC-E) 2009. The localisation algorithm described in Chapter 3 has been tested there, as outlined in Section 7.1.1.

1.7.6 CMRE waterfront

The Centre for Maritime Research and Experimentation of NATO, La Spezia, formerly NURC - NATO Underwater Research Centre regularly host the SAUC-E

Figure 1.13: Somerton Diving Pool. Used to test a localisation algorithm, in preparation for the competition SAUC-E 2009.

Figure 1.14: QinetiQ Ocean Basin Tank, used during the SAUC-E competition 2009.

Figure 1.15: The Centre for Maritime Research and Experimentation of NATO, La Spezia, used for the SAUC-E competition 2010, when localisation with respect to a structure has been tested.

competition. It hosted the Student Autonomous Underwater Challenge - Europe (SAUC-E) from 2010. The structure inspection algorithms described in 4 have been tested there, as outlined also in Section 7.1.1.

1.8 Conclusions

This Chapter has outlined the research objectives and the contribution of this thesis. Then it presented the problem of localisation, formalised with a probabilistic approach. Following that, underwater sensors were outlined, considering both sensors to perceive the environment, and sensors to estimate the motion. Both types will be used in the thesis for localisation. Finally, the different platforms employed as well as the test sites were described. The rest of thesis is organised is the following way:
- Chapter 2 will present the relevant literature for passive localisation approaches, the mathematical background behind Bayesian filtering and some implementations and first tests of standard techniques;
- Chapter 3 will highlight the proposed approaches, and will present the experimental results and comparison with other techniques;
- Chapter 4 will present the topic of navigation around structures, comparing different approaches and showing experimental results;
- Chapter 5 will present the relevant literature for active localisation approaches, highlighting shortfalls of current techniques;

- Chapter 6 will present a novel approach to active localisation, with a full framework proposal to address the topic, presenting experimental results and comparison with other techniques;
- Chapter 7 will present some applications of the presented algorithms, and discuss the presented results, outlining future research paths, which can be undertaken based on the results of this thesis.

Chapter 2

Passive techniques for AUV localisation

2.1 Introduction

The problem of self localising a robot consists in determining its pose in the operating environment, given the observation history, the command history, and the knowledge of the environment. Analytically, it can be expressed as estimating the probability distribution:

$$p(x_t|z_{0:t}, u_{0:t}, m) \tag{2.1}$$

where x_t is the robot position in the environment at time t, $z_{0:t}$ is the sensing history $z_0, z_1, ..., z_t$ up to time t and $u_{0:t}$ is the command history $u_0, u_1, ..., u_t$ up to time t.

This chapter focuses on *passive* techniques for localisation. By the word *passive*, the author means *without the vehicle control in the loop*. It means that the robot tries its best to estimate its state, given the sensing history and the command history. There is no decision making involved, in order to facilitate the localisation process. The module takes the values from the sensors input and gives the best state estimate as output. It is to be noted that some of the related work cited in section 2.2 presents algorithms, which employ *active* techniques. This is due to the different semantic attributed to the adjectives *passive* and *active*. All the work cited in this chapter comply with the above definition of *passive*: no decision making involved. This ambiguity derives from the subjects to which the adjectives are referred. In the case of this thesis, *passive* and *active* are related to the vehicle. As long as there is no decision making, the vehicle is *passive*, whilst when the control is in the loop and a specific decision related to localisation is made, the vehicle is *active*. In other works, the adjectives are referred to devices placed in the environment to facilitate the task of localisation. Therefore, according to those authors, a *passive* localisation approach deals with *passive* devices - for example a cat's eye acoustic buoy, whist an *active* localisation approach deals with *active* devices - for example an active pinger. Those

approach are both classified as *passive* for the scope of this thesis and therefore cited in this Chapter and not in Chapter 5.

The usual sensor used in the underwater world to sense the environment is the sonar, as outlined in section 1.3.3. In addition to sonar values, however, the localisation module could have more information from other sensors, such as compass, altimeter, depth sensor. Those information help to reduce the state space, thus impacting on the efficiency of the algorithms. Additionally, Inertial Measurement Unit (IMU) and Doppler Velocity Log (DVL) can provide estimates of the vehicle motion, giving better estimates than a hydrodynamic model, which would be used with the command history.

This Chapter is organised as follow:

- section 2.2 will present the related work in the field. Several approaches to the localisation problem are presented, with a particular emphasis on the use of filtering techniques and in particular to particle filters;
- section 2.3 will present the Bayesian mathematical background behind this thesis;
- section 2.4 will present the standard Extended Kalman Filter (EKF) algorithm applied for AUV localisation;
- section 2.5 will present some scan matching approaches for AUV localisation.
- section 2.6 will present the Set Membership approach for AUV localisation

Finally, conclusions will summarise the main techniques presented, will present the shortfalls of the presented approaches and will outline the open questions addressed in the following Chapter.

2.2 Related Work

This section will review current approaches in underwater localisation, with a focus on filtering techniques.

2.2.1 Acoustic communication

The earliest method using transponders, put forward by Smith & Cheeseman, uses an Extended Kalman Filter (EKF) to estimate jointly the state of the AUV and the position of the transponders [101]. This approach has been used with success in various cases, for example by Kantor & Singh [52], Kurth *et al.* [62], Djugash *et al.* [25], but the filter suffers from linearisation of non-linear models which can quickly lead to divergence as the covariance estimates become unreliable. In addition, the algorithmic complexity of the EKF algorithm grows quadratically with the number of beacons (in the two dimensional case). For example, the prediction step of the EKF only affects the state of the AUV but involves the computation of the Jacobian of the motion model which also takes into account the transponders.

Several methods have been put forward to reduce the complexity of the EKF algorithm, such as the linear state augmentation principle or the partitioned update approach both presented by Bailey & Durrant-Whyte in [8]. These methods have a linear complexity with the number of beacons.

Scherbatywk analysed the use of a single transponder in a LBL configuration. In addition to successfully demonstrate his approach in simulation, he also pointed out the main disadvantages of this technique: transponders installation, calibration, recovery and possible losses [97].

The theory behind a single beacon LBL approach is well described in the work of Webster *et al.*. The work presented also extends the use of this technique to multiple vehicles. Synchronous clocks on the reference beacon and the vehicles enable the measurement of one-way travel-times, whereby the time of launch of the acoustic signal at the reference beacon is encoded in the acoustic broadcast and the time of arrival of the broadcast is measured by each vehicle. The decentralised navigation algorithm, running independently on each vehicle, is implemented using the information form of the Extended Kalman Filter [115].

Other relevant work using a single beacon for localisation has been presented by Ferreira *et al.* in [32]. To determine its horizontal position, the AUV fuses distances to the single beacon with dead reckoning data, heading and longitudinal velocity. The authors implemented both a Particle Filter and an Extended Kalman Filter and compared the two techniques in terms of performances. The merged solution among Particle Filters and Extended Kalman Filter is however a subset of the solution

proposed in this thesis in section 3.4 and presented in [72].

Willumsen *et al.* additionally showed the feasibility for the AUV to locate itself, with the aid on a single transponder, analysing the efficiency of different approaches, together with quality and quantity of available information [119]. Both active (range) and passive (differential range) measurements achieve a good accuracy. As expected, using both range and differential range gives a quicker convergence in position estimate. All these aids however, need time and movement to obtain good navigation.

For SBL, Storkensen *et al.* proposed to use a support ship equipped with a high-frequency directional emitter able to accurately determine the AUV's position with respect to the mother ship. This approach requires, however, the support of a ship [103], [111]. Furthermore, it cannot be used in many situations as it requires the ship and the vehicle being close to each other. This approach is therefore not suitable for navigation around deep off-shore structures.

Watanabe *et al.* also studied the use of SBL and SSBL (Super Short Base-Line, similar to USBL), analysing its multiple advantages to solve the AUV localisation problem [114], linked to data transmission. It is notable that the proposed approach has been validated at a depth of $1.200m$, with the ROV *Kaiko*.

The French company IXSEA presented a few commercial solutions to fuse USBL with inertial technologies [117]. The GAPS positioning system takes advantage of rigid mounting of INS and USBL, careful synchronising the data coming from an innovative 3D antenna. The coupling of RAMSES and PHINS systems uses not only of ranges and inertial sensors, but also SLAM techniques adapted to subsea positioning needs. Although the article is very commercial-oriented, the presented results for AUV localisation are promising.

Ura and Kim also analyse the fusion among USBL and INS techniques. After the vehicle has reached the maximum altitude for the DVL to be able to track the seabed, the position error is calculated as the difference between SSBL position and corresponding position in INS [109]. The proposed approach has been validated on a in-water mission at Kuroshima Knoll, showing great advantages for the navigation system.

The idea of underwater transponders can evolve into the idea of a proper underwater station, which not only helps the AUV to localise itself, but it can provide tools for recharging and data exchange.The work proposed by Maki *et al.* deals with AUV localisation with respect to a seafloor station [69]. The key idea is acoustic mutual orientation and communication. The AUV firstly sends a signal to the station and then receives the reply with a hydrophone array to estimate direction and distance to the station. On the other hand, the station estimates the direction to the AUV by receiving the signal with a hydrophone array, and sends the information back to the

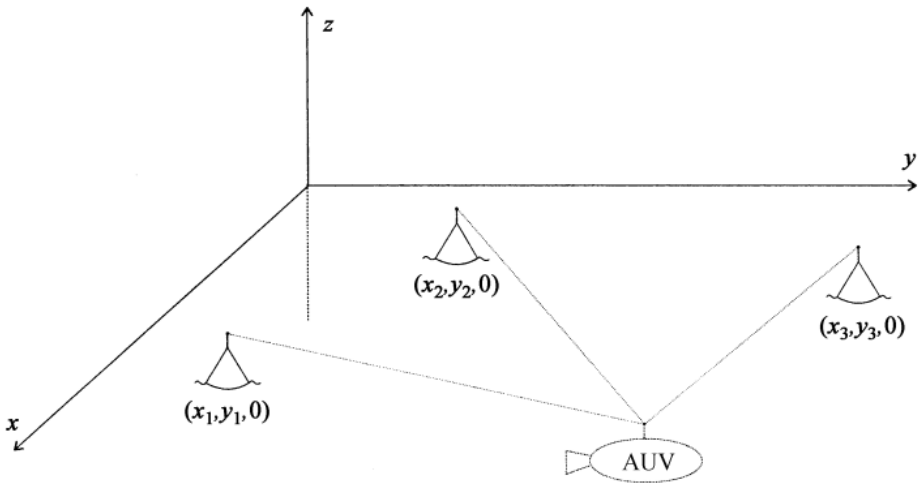


Figure 2.1: Use of freely floating acoustic buoys equipped with GPS connection, communicating with the AUV.

AUV. Then, the AUV can estimate its position and orientation at the station-fixed coordinates without drifts. Furthermore, these measurements are fused with other on-board sensors such as DVL, angular rate gyroscope and depth sensor by particle filter, a probabilistic approach, in order to realise stable positioning robust against sensor noises and lack of measurements.

A different approach on acoustic sensor fusion for AUV navigation was performed by Rigaut *et al.*, showing the advantages of acoustic data fusion, versus a dead reckoning strategy [95]. The matching of the absolute and relative reckoning modules permits to compute a new hypothesis, and to reset the relative estimation in an asynchronous way with the help of the operator.

2.2.2 Communication with surface vehicles/buoys

Having the possibility to use the GPS would be however very beneficial, so some other approaches have been explored, arriving to the so-called *Surface-LBL*. This approach uses third entities on the surface, and the submerged vehicle acoustically communicate with the surface. Caiti *et al.* developed an acoustic localisation technique using freely floating acoustic buoys equipped with GPS connection [15]. This system requires the buoys to emit a ping at regular time intervals with the coded information of its GPS position. The vehicle can locate itself using time-of-flight measurements of acoustic pings from each buoy. Figure 2.1 shows the proposed framework. The limitations of this approach are the necessity to deploy enough floating buoys in the mission area, the need to collect them after the end of the mission and a non efficient communication scheme, as the buoys periodically send acoustic messages. Limitations

for deep water missions are also evident.

A similar approach is presented by Yang *et al.*, using three surface buoys or vessels [120]. Acoustic ranges are calculated using the equivalent sound speed profile and travel times of sound rays. The linear distances can be obtained through ray tracing and the position thus calculated through geometric calculation. Simulation results have shown an accuracy of $20cm$ in the area covered by the three transducers at a depth of $500m$.

Yang *et al.* also proposed a sonar-based approach for passive localisation of an AUV [121]. It is however to be noted that the term *passive* is used with a different meaning than the one presented in this thesis, and in particular in this chapter. Yang refers with passive localisation to techniques where passive receivers listen for a signal broadcasted by the AUV and then infer its position. Once again, in the scope of this thesis *passive localisation* means that the vehicle has no way to actively link the control loop and the robot actions with the localisation process, which only processes the data produced by the sensors.

2.2.3 Terrain-based navigation

Terrain-based navigation is another very important area for AUV localisation. The estimated position is not calculated trying to find an acoustic fix either with transponders or with floating buoys, but it is relative to the terrain. The AUV measures the topography of the bottom using on-board sensors, and correlates those measurements with an existing bathymetry map in order to estimate its position on this map during the mission, as shown in Figure 2.2. The same concept can be extended for navigation around underwater structures. The underlying idea is the use of sensors to perceive the environment and to match a previously known map, being it a bathymetry map or a map of the environment with structures and objects.

This produces a non-trivial advantage in terms of operational cost (including time and money) avoided for the deployment of the aiding devices. Terrain-based navigation has been widely used for decades in aircraft and cruising missiles. An example is in the early work of Hostetler and Andreas, where they explored the application of non-linear Kalman filtering techniques to radar terrain-based navigation [41].

One of the first works in the underwater domain is presented by Newman and Durrant-Whyte [81]. Analysing the nature of the sonar beam, they successfully estimated the ocean floor gradient, as a world feature. Applying a simple target extraction algorithm, the AUV was able to identify and rely on natural features for navigation.

Nakatani *et al.* proposed a terrain-based approach using a Particle Filter for stochastic estimation [110]. The approach has been successfully validated through sea experiments and the work pointed the way to active techniques, which are the

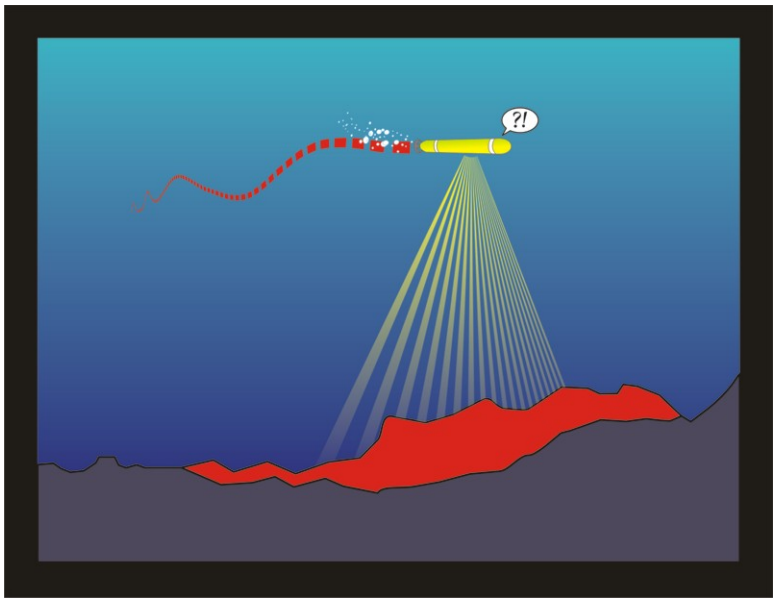


Figure 2.2: Terrain-based navigation: the AUV measures the topography of the bottom and estimates its location in the map.

core of Chapter 5 and 6.

The work presented by Sarma represents an important contribution in the field of map-matching based self-localisation. The AUV is equipped with a multi-beam high frequency sonar and matches the received data from the sensor with a bathymetry map, *a priori* known. The estimated position is computed using the Maximum Likelihood Estimator, widely used for land robotics, for example in [85], [104], [68], [82], [42].

Carreno *et al.* presents several works using Particle Filters for terrain-based Navigation [18]. Karlsson *et al.* proposed a particle filter approach for terrain-based AUV navigation, based on a SIR-Particle Filter. An INS was used for measuring the robot displacement and an echo sounder was employed for measuring the altitude. The focus was however more on the mapping problem than on the localisation one [53].

An analysis of particle filters for terrain-based navigation in presence of tides is given by Anonsen & Hallingstad, who also compared standard Monte Carlo approaches with a Bayesian Point-Mass Filter (PMF) [5]. Although results of PMF were slightly better, the computational needs explode when going into an estimation of a 3D state, whereas it still remains tractable for particle filters.

Silver *et al.* presented a particle filter merged with scan matching techniques [99]. They use a particle filter and an approximation of the likelihood of sensor readings, based on nearest neighbour distances, to approximate the probability distribution over possible poses. The initial robot location is not considered as an issue in this work, as it focuses mainly on trajectory tracking and mapping.

The choice of sensors always play a major role, due to limited payload capability of AUVs. As many survey-class AUV are equipped with a sidescan sonar, to acquire images of the seabed, Zerr *et al.* investigated the use of the same sensor data to improve navigation [124]. It is assumed that the AUV surveys a known area, so a previous map of the environment is known and available for pre-processing and feature extraction. Matching the results of real time payload data processing with a knowledge database has demonstrated potential aid to AUV navigation when performing routine missions.

Huang *et al.* proposed a novel inertial-SLAM algorithm for AUVs, by fusing IMU's data with information from the sonar [43]. The AUV can estimate the velocity and pose by Inertial-SLAM only using on-board IMUs and sonar sensors with. According to the simulation results, the proposed system is more efficient and accurate than a standard EKF-SLAM.

An extension to terrain-based navigation is represented by the work of Kimball *et al.* [56]. Their proposed approach can be used for AUV navigation relative to an iceberg. Navigation techniques based on inertial sensors or GPS cannot provide ice-relative navigation accounting for the full motion of free-floating icebergs, especially for iceberg rotation. The proposed approach has been validated post-processing multi-beam sonar data, acquired by a ship circumnavigating the iceberg. The unknown movement of the iceberg introduces a higher level of uncertainty than standard terrain-based navigation. It is however to be noticed that a transposition from a side-looking sonar mounted on a ship to a fully autonomous vehicle exploring under-ice would not be an easy task and would certainly bring new challenges.

The idea of terrain-based navigation is similar to the one explored in this thesis and presented in this chapter, though the proposed approach of this thesis does not deal mainly with explicit feature extraction and with data association.

The work presented by Kondo *et al.* is in the same direction of this thesis, dealing with navigation around underwater structures [58]. The proposed solution employs a Particle Filter for global localisation and position tracking, while the planner issues waypoints once a convergence has been reached.

2.2.4 Magnetic navigation

Another field explored in subsea navigation is related to magnetic fields. Researchers have either explored how a strong magnetic field can influence navigation, or - on the other hand - can help the localisation process.

Kondo and Ura analysed the difficulties for an AUV of correctly localising itself in presence of strong magnetic disturbances, which are typical near steel underwater structures, or in presence of a particular geological configuration [59]. To prevent

the navigation being affected by the disturbances, they proposed a relative approach, based on images by a CCD camera and laser pointers. Both cameras and laser pointers can work only in clear water, so this approach is not transferable to deep water application without any change. However, changing the main external sensor to sonar, and navigating around the structure with visual servoing techniques allow the vehicle to prevent unwanted jumps in the navigation state (and thus in the control) due to magnetic disturbances.

Magnetic disturbances have been research subjects also for Huang and Hao, [123]. The dipole magnetic anomaly caused by ferromagnetic object or geologic structural change, mixed with geomagnetic field has been investigated and the effect for AUV localisation aided by geomagnetic anomaly. A novel localisation algorithm is presented and validated in simulation.

Magnetic fields and magnetic-aided localisation technique are out of the scope of this thesis, but they are mentioned for completeness on the different localisation techniques.

2.2.5 Other approaches for localisation

Localisation in Partially Known Map

The problem of localisation usually assumes a complete knowledge of the environment. Not many publications deal with localisation in partially known maps, while it is a very important topic, due to unforeseeable obstacles in the sensor field of view, and discrepancies which might occur between a map of underwater structures and their real position and orientation after years, especially if in presence of strong currents. Cristi analysed this problem presenting tank results of an AUV successfully locating itself also in presence of unexpected objects [23]. The approach chosen by Cristi was the use of neural networks to model potential functions for the environment. The approach chosen in this thesis, and described in section 3.3.1 deals mainly with Particle Filters and analyses their ability to handle unknown objects in the environment.

Underwater Sensor Networks

Luo *et al.* presented a localisation system for Underwater Sensor Networks (UWSN), with the use of an AUV [67]. Based on directional signals, which are transmitted by an autonomous underwater vehicle (AUV), the system is able to correctly estimate the node positions. The AUV is considered as a precise source of localisation information, with bounded error. This problem can be mirrored, with uncertain AUV position and with active beacons, as explained by Petillot *et al.* in [90].

Other uses for particle filters

In most of the presented approaches, particle filters are used to estimate the state of the robot. In some mirror scenarios, they were employed to estimate a beacon, or the state of another vehicle. Their use in robotics and in the underwater domain is not however restrained to vehicle localisation only, but can be used to estimate other mission-dependent variables. A work from Ortiz *et al.* for example addresses the use of particle filters for tracking undersea narrow telecommunication cables [88].

Simultaneous Localisation and Mapping

The problem of localisation can be solved together with the mapping problem, thus arriving to a full Simultaneous Localisation and Mapping (SLAM) approach. Although solving the full SLAM problem is out of the scope of this thesis, a very brief review of relevant work in the underwater domain is given. Feder *et al.* explored the issue of long-term performance of SLAM for an AUV equipped with a forward looking sonar [31].

The underwater group at the Australian Centre for Field Robotics (ACFR) is a leading group in underwater SLAM. In the work presented by Williams *et al.*, a stereo-visual SLAM is presented in order to estimate the vehicle position during survey tasks [118]. The presented results of the mapping, obtained by data post-processing, show the capabilities of the algorithm proposed. Barkby *et al.* propose a featureless approach using an a-priori low-resolution map, in order to enforce consistency between the prior map and the AUV bathymetry [9].

Wang *et al.* present a new SLAM algorithm based on support vector machines(SVM) adaptive Extended Kalman Filter(EKF) for autonomous underwater vehicle(AUV) to reduce the influence of the change of statistical characteristics of the system noise and the observe noise [48].

Li *et al.* propose a range sonar array based SLAM method [65]. The vehicle's operational environment is a water tank whose dimension are known to the AUV. Aided by the knowledge of the general environment, the EKF-based SLAM is able to map unknown objects in the tank.

Other solutions to the general SLAM problem are the Extended Information Filter [108], the Unscented Kalman Filter, presented by Julier & Uhlmann, in [49] and the FastSLAM algorithms, developed by Montemerlo *et al.* [77].

Clark *et al.* applied Particle-Filter based FastSLAM for data post-processing collected by a small ROV for archaeology purposes [20].

2.2.6 Critical Analysis

This section has presented several approaches for AUV localisation and several approaches of the use of filtering techniques, with a special focus on Particle Filters. Relying only on state sensors, like for example DVL and compass, the state estimation diverges over time from the real trajectory. In order to correct the estimation, there are two main approaches, both based on acoustic:

- **use of acoustic beacons**: the vehicle communicates with the beacons to compute its location;
- use of sensor data: the vehicle observes the environment (either the seabed, or underwater structures) and matches the sensor data with a previously known map.

This chapter will focus on the second approach. Whilst the first one is already routinely used, it does have disadvantages linked to deployment and recovery. The second approach is certainly more challenging, and requires a high level of autonomy in the robotic systems. The goal of this research is to present a generic system which does not require any external support, like acoustic beacons. The interest is mainly on increasing the capabilities of the robot, on the adaptation of localisation techniques for the underwater world, and on some optimisations. A key interest is in probabilistic approaches, as they can address uncertainty typical of real world robotic tests, with uncertainty in process model, sensor readings, etc. Therefore in the next section a Bayesian formulation is presented, alongside with the mostly used techniques which approximate the Bayesian formulation. Thereafter, several proposed approaches are presented.

2.3 Bayesian filtering framework

2.3.1 Optimal Bayesian filtering

The state of the AUV at time t is a random vector denoted by X_t that usually includes the position and the speed of the AUV. The evolution of the AUV's state is provided by the motion model. Usually, the motion model assumes that (X_k) is a Markov process entirely defined by the transition kernel $p(x_k|x_{k-1})$ and the distribution of the initial state X_0. The transition kernel is related to the measurements provided by the IMU or the DVL that predict the future state of the AUV given its past state. The state X_t is not directly observed but is known through measurements regularly gathered by the AUV. The measurements are, for example, the scan of a

rotating sonar or range measurements with respect to acoustic transponders. The measurement at time t is a random vector denoted by Y_t that only depends on X_t. The measurement model provides the conditional density $p(y_t|x_t)$ that statistically links the observation Y_t to the state X_t. These assumptions are those of a hidden Markov model.

The objective in Bayesian filtering is to calculate the conditional density $p(x_t|y_{0:t})$ so that one has an estimate of the state X_t through the conditional expectation

$$\mathbb{E}[X_t|Y_{0:t}] = \int x_t p(x_t|y_{0:t}) dx_t \tag{2.2}$$

This conditional expectation is the estimate of X_t given $Y_{0:t}$ that minimises the mean square error, namely

$$\mathbb{E}[\,|X_t - \mathbb{E}[X_t|Y_{0:t}]|^2\,] \leq \mathbb{E}[\,|X_t - \psi(Y_{0:t})|^2\,] \tag{2.3}$$

for any other estimate $\psi(Y_{0:t})$ of X_t given the observations $Y_{0:t}$. The conditional density $p(x_t|y_{0:t})$ may be recursively calculated from $p(x_{t-1}|y_{0:t-1})$ through the following exact prediction and correction steps

$$p(x_t|y_{0:t-1}) = \int p(x_{t-1}|y_{0:t-1}) \underbrace{p(x_t|x_{t-1})}_{\text{Transition kernel}} dx_{t-1} \tag{2.4}$$

and

$$p(x_t|y_{0:t}) = c_t\, p(x_t|y_{0:t-1}) \underbrace{p(y_t|x_t)}_{\text{Measurement model}} \tag{2.5}$$

The normalisation constant c_t is unknown and given by

$$1/c_t = \int p(x_t|y_{0:t}) dx_t = \int p(x_t|y_{0:t-1}) p(y_t|x_t) dx_t \tag{2.6}$$

Assuming that the motion and the measurement models are both linear and Gaussian, the distribution of $X_t|Y_{0:t}$ is also Gaussian. The mean and the covariance matrix of $X_t|Y_{0:t}$ may then be calculated using the recurrence relations of the Kalman filter [50, 51]. The calculation of the conditional expectation $\mathbb{E}[X_t|Y_{0:t}]$ is exact so that the Kalman filter is optimal, namely it provides the estimate of X_t given $Y_{0:t}$ that minimises the mean square error, which was mentioned previously.

The motion and the measurement models of an AUV are seldom both linear and Gaussian. One has consequently to provide approximate and thereby suboptimal solutions to the non-linear non-Gaussian Bayesian filtering problem. Various approximation schemes have been put forward among which are the Monte Carlo and the

sum of Gaussian approximations. The former approximation is at the core of any particle filter and the latter is known as the Gaussian Sum Filter.

2.3.2 Kalman Filter

The Kalman Filter (KF) [50], [116] is a filter that can be derived directly by the optimal Bayesian filtering under the assumptions that the system is linear and the noise is Gaussian. Under this linear assumption the system can be described by

$$\begin{aligned} x_{t+1} &= F_t x_t + w_t \\ z_{t+1} &= H_t x_t + v_t \end{aligned} \tag{2.7}$$

The system noise $w_t \approx \mathcal{N}(0, \sum_{w_t})$ and the observation noise $v_t \approx \mathcal{N}(0, \sum_{v_t})$ are zero mean normally distributed. The key advantage of the Kalman Filter is that it represents the distributions in closed form, in terms of means and covariance matrix. The update of the Kalman filter can be carried out in the time of a matrix multiplication ($O(n^3)$, where n is the state dimension).

The iterative algorithm of the filter is the following:

- predict:

$$\begin{aligned} x_{t+1}^{'} &= F_t x_t \\ \Sigma_{t+1}^{'} &= F_t \Sigma_k F_t^T + \Sigma_{w_t} \end{aligned} \tag{2.8}$$

- update:

$$\begin{aligned} K_k &= \Sigma_t^{'} H_t (H_t \Sigma_t^{'} H_t^T + \Sigma_{v_t})^{-1} \\ x_k &= x_t^{'} + K(z_t - H_t x_t^{'}) \\ \Sigma_t &= (I - K_t H_t) \Sigma_t^{'} \end{aligned} \tag{2.9}$$

Unfortunately, in underwater robot domain, the evolution model, as well as the observation model are non linear, and the noise cannot be considered Gaussian. However, for mild evolution laws and some specific problem, a non linear extension can be used: the Extended Kalman Filter (EKF) [116], in which the functions f and h can be non linear, but the covariance matrix is calculated in a local linearisation under the current state. The extended Kalman filter algorithm can be expressed as

- predict:

$$
\begin{aligned}
x^{'}_{t+1} &= f_t x_t \\
\Sigma^{'}_{t+1} &= F_t \Sigma_k F_t^T + \Sigma_{w_t}
\end{aligned}
\tag{2.10}
$$

- update:

$$
\begin{aligned}
K_k &= \Sigma^{'}_t H_t (H_t \Sigma^{'}_t H_t^T + \Sigma_{v_t})^{-1} \\
x_k &= x^{'}_t + K(z_t - h_t x^{'}_t) \\
\Sigma_t &= (I - K_t H_t)\Sigma^{'}_t
\end{aligned}
\tag{2.11}
$$

where $F_t = \nabla_x f_t|_{x_t}$ and $H_t = \nabla_x h_t|_{x_t}$

The key limitations in the use of extended Kalman filter lies in the strong assumptions that have to be done on the estimated system, namely: Gaussian noise, and linearisability. In most of the robotic systems used for localisation and SLAM the uncertainty is not expressible as a Gaussian distribution, being multi modal and non regularly shaped. When more modes are present in a distribution, dealing with multiple hypotheses is needed, while the Kalman Filter works on their mean. In these situations its use is prone to failure. Moreover, the linearisation of the system can introduce some systematic error in the estimate. Finally, some systems cannot be linearised (being their 1st order derivatives null), thus the Extended Kalman Filter cannot be applied. In these contexts a second order extension to the Kalman filter: the Unscented Kalman Filter (UKF) has been proposed in [112]. While the UKF in general behaves better than the Kalman filter, the hypotheses of Gaussian noise is still required to hold. Nevertheless, the Kalman Filter is one of the most used tools in localisation and SLAM, due to its simplicity. Moreover, when the underlying hypotheses hold, it exhibits a strong convergence rate if compared with other filtering techniques [112].

2.3.3 Particle filters

Any particle filter relies on a Monte Carlo approximation of the conditional density $p(x_t|y_{0:t})$ using a finite set of N points ξ_t^i in the state space called *particles*. The approximation is of the form

$$
p(x_t|y_{0:t}) \simeq \sum_{i=1}^{N} w_t^i \delta_{\xi_t^i}(x_t) \quad \text{where} \quad \sum_{i=1}^{N} w_t^i = 1 \tag{2.12}
$$

and where $\delta_{\xi_t^i}$ denotes the usual Dirac function, namely

$$\delta_{\xi_t^i}(x) = \begin{cases} 1 & \text{if} \quad x = \xi_t^i \\ 0 & \text{otherwise.} \end{cases} \tag{2.13}$$

Formula (2.12) may be interpreted in the following way: the denser the particles in a region of the state space and the higher their weights, the higher the probability that the state lies in this region. Assuming that one knows how to sample from $p(x_t|y_{0:t})$, the Monte Carlo approximation becomes

$$p(x_t|y_{0:t}) \simeq \sum_{i=1}^{N} \frac{1}{N} \delta_{\xi_t^i}(x_t) \quad \text{with} \quad \xi_t^i \sim p(x_t|y_{0:t}) \tag{2.14}$$

The previous assumption is unrealistic since it implies that the Bayesian filtering problem is solved. Any particle filter rely on the importance sampling principle which provides a mechanism to build a Monte Carlo approximation of $p(x_t|y_{0:t})$.

Suppose one would like to sample from a distribution whose density f is of the form $f(x) = c\, r(x) g(x)$ where

1. g is the density of a distribution from which it is easy to sample and which is called the proposal distribution;

2. r is a weighting function easy to evaluate;

3. $c = \int f(x)dx$ is an unknown normalisation constant.

The importance sampling principle provides the following approximation for f.

$$f(x) \simeq \sum_{i=1}^{N} w_i \, \delta_{\xi_i}(x) \quad \text{with} \quad \xi_t^i \sim g(x) \tag{2.15}$$

where the weights $w_i = f(\xi_i) / \sum_{j=1}^{N} f(\xi_j)$ are not equal to $1/N$ since they account for the particles being generated using another distribution than f.

The first particle filter ever proposed was the *bootstrap filter* introduced in [37] but the most commonly used particle filter is the *sampling with importance resampling* (SIR) filter. Other kinds of particle filters such as the auxiliary particle filter or the kernel filter have been introduced in order to improve the performance of the SIR algorithm. The interested reader can found an overview of the existing sequential Monte Carlo methods for Bayesian filtering in [16, 3].

The SIR filter intends to reproduce the optimal prediction and correction steps of the optimal Bayesian filter. The three steps of iteration $t \geq 1$ of the SIR algorithm are as follows.

1. **Selection:** Generate $\tau_t^i \sim (w_{t-1}^1, \ldots, w_{t-1}^N)$
2. **Propagation:** Generate $\xi_t^i \sim p(x_t|\xi_{t-1}^{\tau_t^i})$
3. **Correction:** Set $w_t^i \propto p(y_t|\xi_t^i)$

In the SIR particle filter, the propagation step of iteration t uses the transition kernel to simulate the new set of particles $(\xi_t^1, \ldots, \xi_t^N)$, as in the bootstrap filter. However, only the most likely particles at time $t-1$ are selected to generate the particles at time t. The selection is made according to the weights since the higher a weight, the higher the corresponding particle is in adequacy with the observations. This selection step makes the SIR particle filter more efficient than the bootstrap filter.

2.3.4 Gaussian Sum Filter

Particle filters rely on a Monte Carlo approximation of the distribution of $X_t|Y_{0:t}$. They require a large number of particles so that the approximation is accurate enough and thereby have a high computational cost. A long time before the emergence of particle filters, an approximation scheme based on a mixture of Gaussian distributions was suggested in [2]. This Gaussian sum approximation was shown to perform better than the Extended Kalman Filter while being compatible with the computational capabilities of that time. The conditional density $p(x_t|y_{0:t})$ is approximated by

$$p(x_t|y_{0:t}) \simeq \sum_{i=1}^{N} \alpha_t^i \Gamma(x_t\,;\, m_t^i, P_t^i) \quad \text{with} \sum_{i=1}^{N} \alpha_t^i = 1 \tag{2.16}$$

where $x \mapsto \Gamma(x\,;\, m, P)$ denotes the probability density function of a Gaussian multivariate distribution of mean m and covariance matrix P. The individual means and covariance matrices m_t^i and P_t^i are updated according to the recurrence relations of the Extended Kalman Filter. The interested reader can found a complete description of the Gaussian Sum Filter in [2].

2.3.5 Bayesian approach to the SLAM problem

The Bayesian filtering framework is suitable to estimate the state X_t of an AUV. The Simultaneous Localisation and Mapping (SLAM) problem also requires the estimation of the state X_t along with the estimation of the map of the environment. In the SLAM problem considered in this section, the environment is composed of acoustic transponders whose positions are to be found. The position of each transponder is denoted by B_i and B denotes the set of all the transponders.

The Bayesian approach to the SLAM problem requires the calculation of the conditional density $p(x_t, b|y_{0:t})$ so that one has an estimate of (X_t, B) through the conditional expectation

$$\mathbb{E}[X_t, B|Y_{0:t}] = \int p(x_t, b)p(x_t, b|y_{0:t})d(x_t, b) \tag{2.17}$$

2.4 Extended Kalman Filter for Localisation

In this section a detailed presentation of the EKF filter is presented, together with numeric results of an implementation of the filter. The map consists in $m_1, ..., m_K$ object (landmark) locations, the state vector is the location vector of the robot, and the observation vector consists in the landmark locations, seen in the reference frame local to the robot. If the landmarks can be uniquely identified, then the solution to the localisation problem is relatively straightforward and this method works well, but if the data association cannot be solved it is necessary to adopt multiple hypotheses tracking techniques that overcome the mono-modality of the Kalman Filter. Another way of using the Kalman Filter for localisation is to use some forward observation model $p(x|z)$ that returns the set of feasible positions that are compatible with the current observation. For example in [39], [21], the location of the robot is evaluated by map matching techniques among the proximity data ad a geometric map, then used to feed the KF. In particular, in [44, 38] an efficient matching method in the Hough parameter space is used, for estimating the position given the observation. The main limitation of this technique lies in the requirement of unique landmarks, or in the presence of some unambiguous forward observation model. Unfortunately, the use of unambiguous models is impossible in environments presenting symmetries.

Two types of landmarks can be analysed, artificial and natural landmarks. The artificial are placed for the purpose of being landmarks, examples include bar codes and floor stripes. The natural landmarks are extracted from the environment without any change being made. In the work presented in this section, artificial landmarks are defined as narrow vertical pillars placed on the bottom surface of the underwater environment. This has the advantage of providing arbitrarily detailed information including its pose in the global map frame.

2.4.1 Data Association

Feature Matching or Date Association is implemented using Individual Compatibility Nearest Neighbour (ICNN) Method.

- Take a measurement z_k with uncertainty R_k

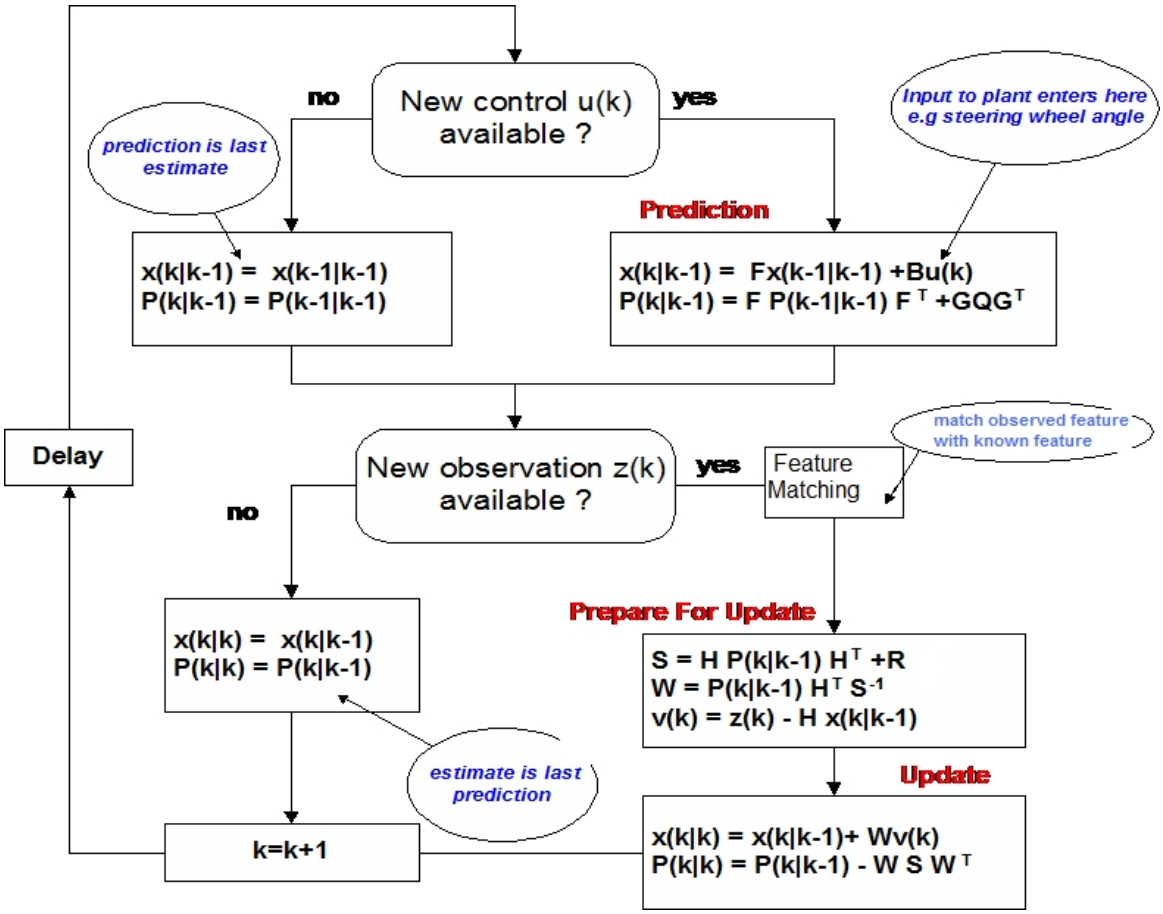


Figure 2.3: A complete picture of EKF localisation for artificially defined landmarks

- For every map feature x_j:
  - Compute the robot related position of the feature
    $[h_j(x_k^-), H_j P_k^- H_j^T]$
  - Compute the Innovation and its uncertainty
    $v_{kj} = z_k - h_j(x_k^-)$
    $S_{kj} = H_j P_k^- H_j^T + R$
  - Compute the mahalanobis distance
    $D_{kj}^2 = v_{kj}^T S_{kj}^{-1} v_{kj}$
  - Does it Pass compatibility test?
    $D_{kj}^2 < \chi_{d,\alpha}^2$
- Select the Compatible feature with the smallest distance

The Block Diagram in Figure 2.3 shows the actual implementation of the EKF localisation for underwater environment defined by forest of narrow pillars. A 3 DOF vehicle navigates in the map in a 2D trajectory defined a priori.

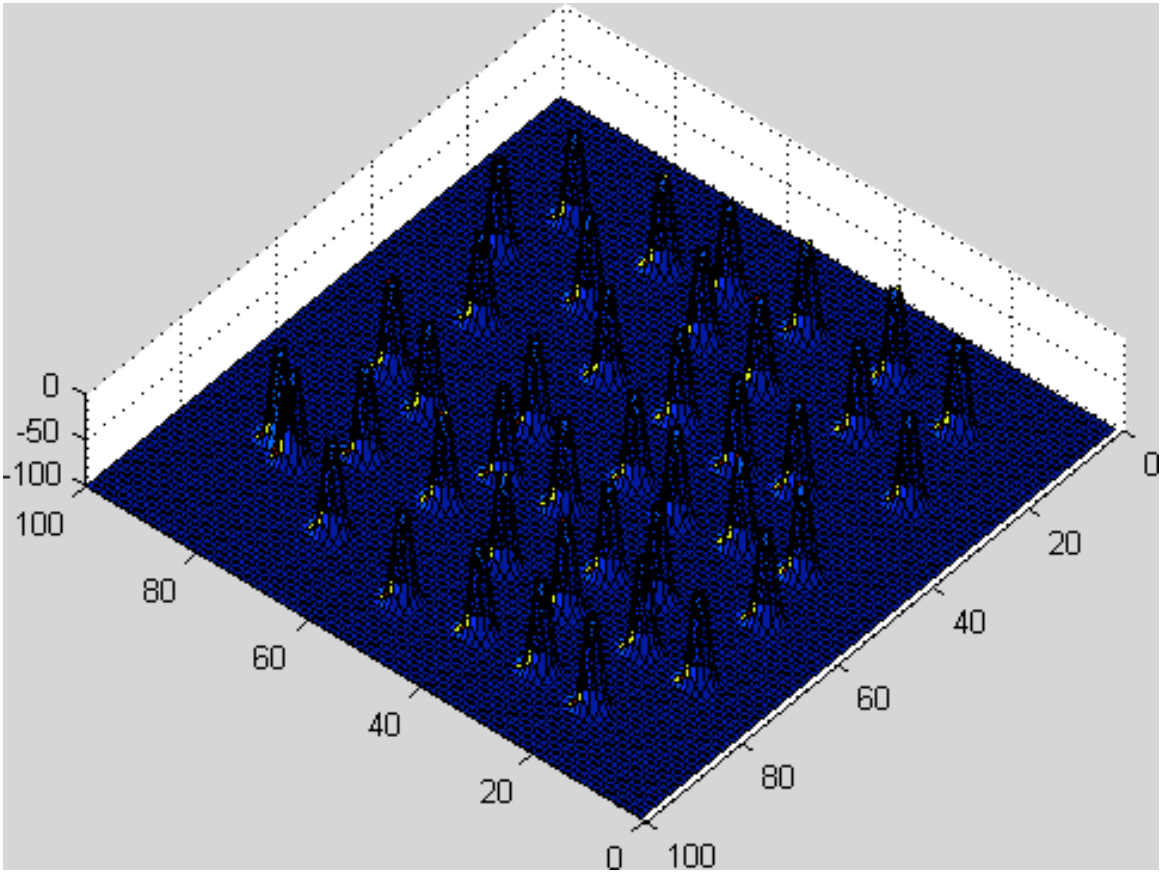


Figure 2.4: A simulated map containing landmarks with known global position.

2.4.2 Numeric results

The Matlab experimental platform is a 3-DOF Autonomous Underwater Vehicle (AUV). The vehicle navigates in a 3D trajectory defined in the *a-priori* map. The vehicle is equipped with a single 2D forward looking sonar. The sonar head is always assumed to be aligned with the heading angle of the vehicle. The maximum range measurement of the sonar is limited to 100 meters and its angular field of view is limited to 90 deg. The map is represented by a number of narrow long pillars placed at the bottom of the underwater environment. Figure 2.4 shows this map. Figure 2.5 and Figure 2.6 shows the localisation results using the motion model only, that is, without the update model of Kalman Filter. Even though the initial state of the vehicle is known, after the first 50 iterations, the estimated trajectory diverts from the ground truth. This implies the vehicle will incur a significant mislocalisation after 50 to 100 time steps, according to 100 tests. Figure 2.6 further shows the sub-linear increase of uncertainty and error in the x, y and θ degree of freedoms. It is also shown to grow rapidly as the vehicle keeps navigating. The increase in error and covariance in the vehicle state, however, can be corrected using EKF update model. In the update model of EKF, the distance from the predicted position to each of the known

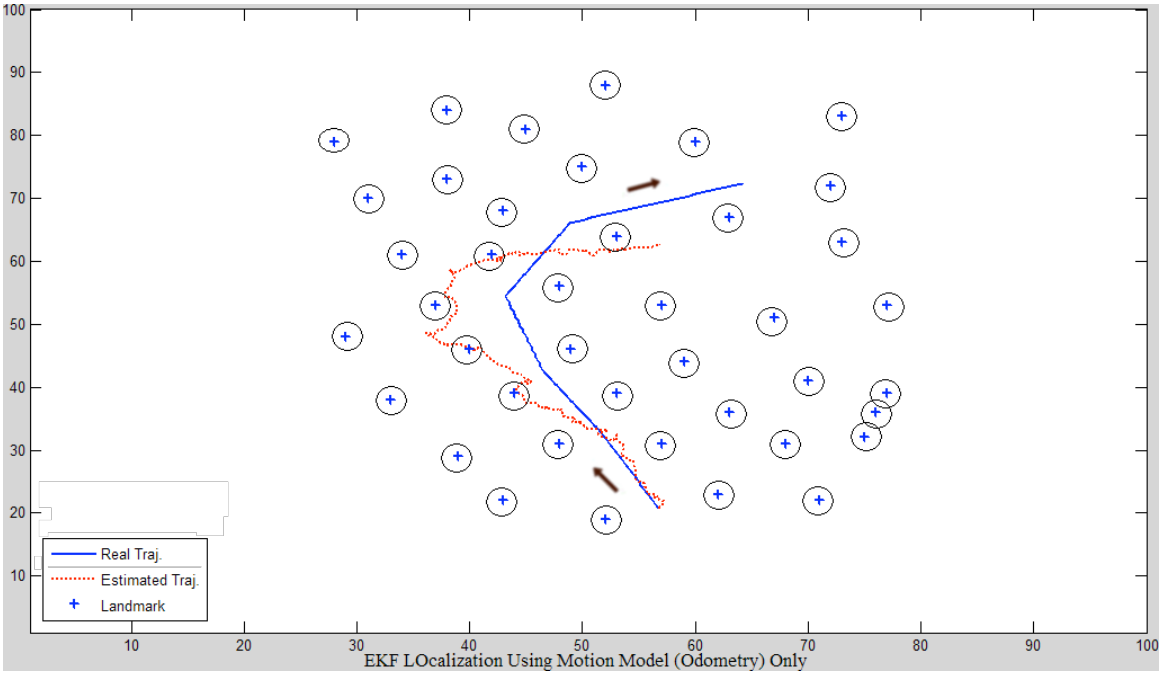


Figure 2.5: EKF-Prediction Model Localisation, Real and Estimated Trajectory.

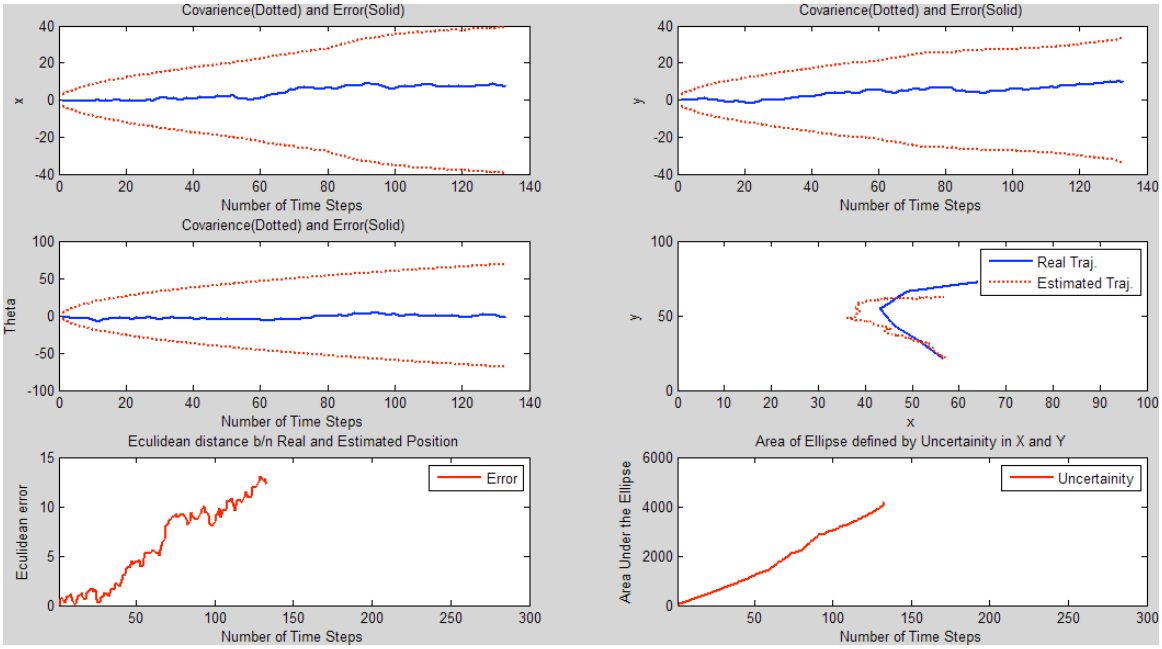


Figure 2.6: EKF-Prediction Model Localisation, Error and Uncertainty Plots.

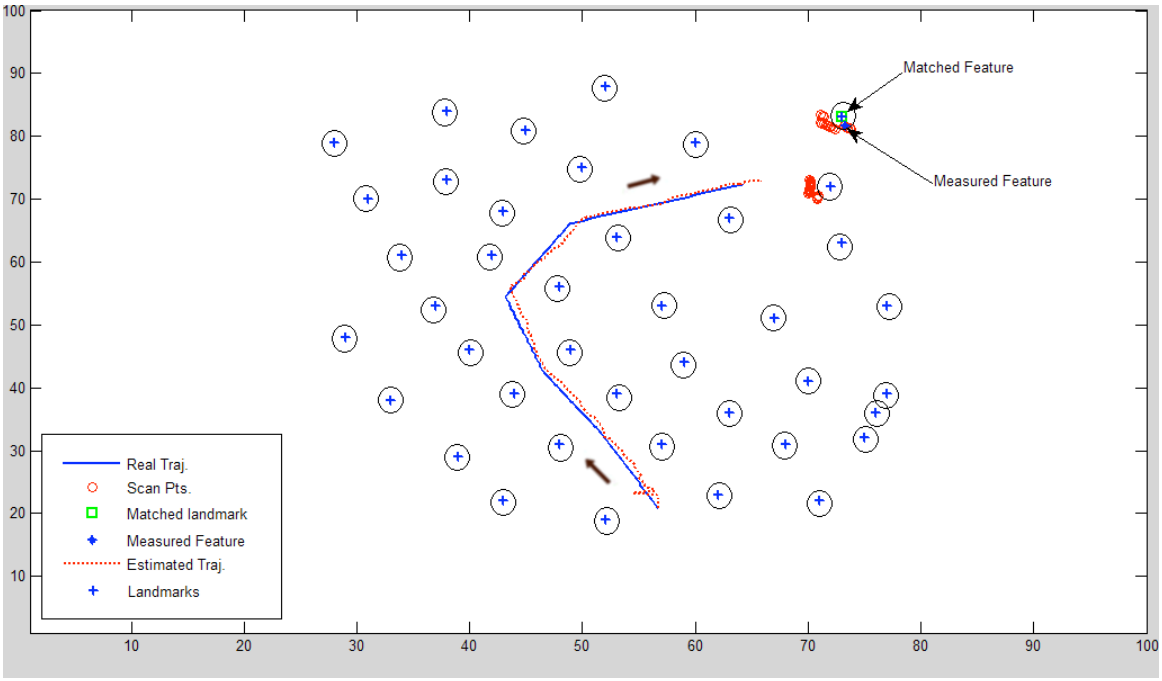


Figure 2.7: EKF-Update Model Localisation, Real and Estimated Trajectory.

landmarks is computed and a feature matching is performed to match the observed landmarks to one of the known features in the map. Consequently, the innovation and covariance between this landmark and real measurement to a feature is computed. The distance in the feature matching stage is computed in the Mahalanobis sense. Finally, the Kalman gain is computed from the innovation covariance and uncertainty of the prediction stage (i.e. motion model) and the state of the vehicle is consequently corrected. The experimental results for the update model are shown in Figure 2.7 and Figure 2.8. As can be seen from the above figure, for the first movement, the error between the real and estimated position is very high. This is due to the fact that the vehicle performs the first iteration without taking a measurement and the localisation is performed only from the information of the motion model. For iterations greater than two there is a fairly small and decreasing error. In addition, during the tests the initial position of the vehicle is assumed to be known. The uncertainty in the state of the vehicle is also estimated and plotted and it is shown to decrease as the vehicle navigates.

2.5 Scan Matching for localisation

Scan matching techniques can also be used for robot localisation. Starting with two sets of range readings and an initial guess for the displacement between them, the algorithm iteratively refines the displacement estimation by generating pairs of

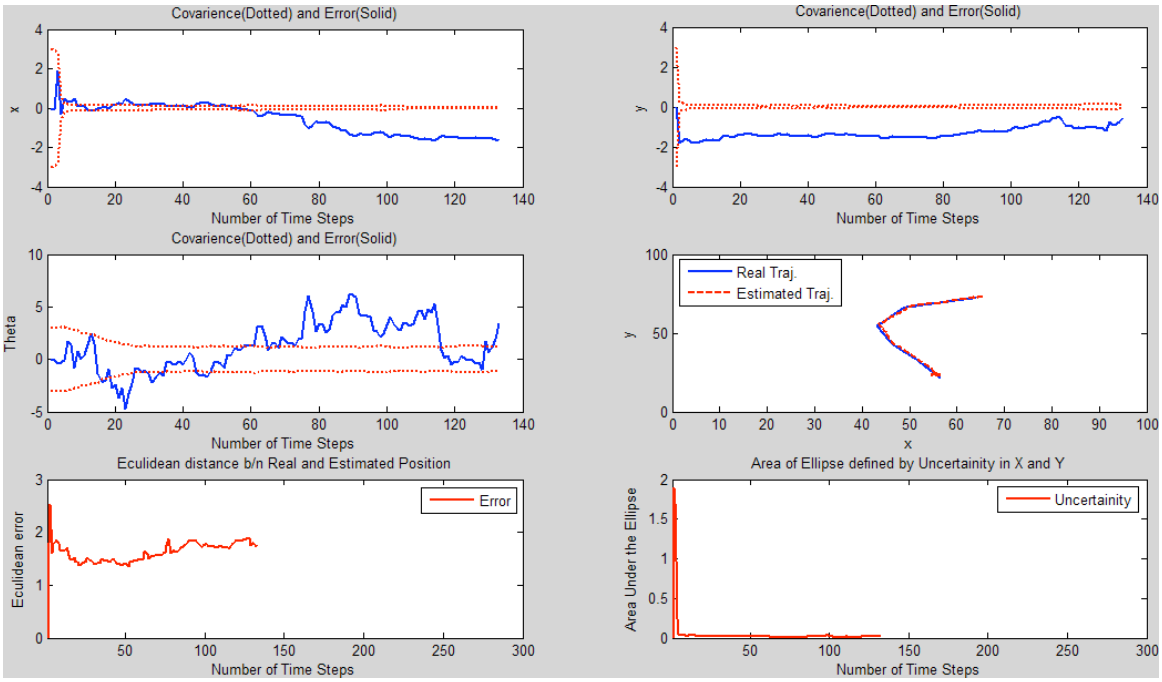


Figure 2.8: EKF-Update Model Error and Uncertainty Plots.

corresponding points on the scans and minimizing an error metric. The most popular methods to perform scan matching is the Iterative Closest Point (ICP) [11], [96], Iterative Dual Correspondence (IDC) [66] or the Metric-Based Iterative Closest Point (mbICP) [76]. However none of these methods take into account sensor noise, which is quite common in underwater sonar. Two types of Scan Matching frameworks are reviewed. Numeric results as well as results from the post-processing of sonar data are presented.

2.5.1 Probabilistic Sonar Iterative Correspondence (psIC)

Probabilistic Sonar Iterative Correspondence (psIC) is a sonar scan matching framework where both the sparseness and noise of the readings are taken into account [78]. This is accomplished by means of probabilistic models of ultrasonic and odometric sensors as well as a method to propagate the error through them. The matching process is accomplished considering the Mahalanobis distance.

Error Models

The two scans used in scan matching are named *current scan* (S_{curr}) and *reference scan* (S_{ref}), being the current scan the most recently gathered scan, and the reference scan a previously gathered scan. These two scans are then grouped by using the odometric information of the vehicle. The travelled distance has to be long enough to acquire a large scan. In practice travelled distance between one and two meters

is a good choice. By performing the grouping process, the resulting scan is subject to two sources of error. Uncertainty in the range and bearing readings and grouping error (i.e. from odometry error).

A sonar reading taken by sonar sensor s at time j is assumed to be a Gaussian random variable, $x_j^s = \mathcal{N}(\hat{x}_j, P_j^s)$. The mean $\hat{x}_j$ represents the translation and rotation of the reference frame located at the sonar reading coordinate, with respect to the reference frame s, located at the sensor position and aligned with the axis of the sonar beam. Thus, $\hat{x}_j = [x, y, \theta]^T$ and P_j^s is a covariance matrix of the form

$$P_j^s = \begin{pmatrix} \sigma_{xx}^2 & 0 & 0 \\ 0 & \sigma_{yy}^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{pmatrix} \tag{2.18}$$

Where σ_{xx} and σ_{yy} model the range and σ_θ the angular uncertainty. It has been experimentally set as

$$\begin{aligned} \sigma_{xx} &= \frac{r}{100}\cos(\theta) \\ \sigma_{yy} &= \frac{r}{100}\sin(\theta) \\ \sigma_\theta &= \frac{r}{2}\tan(\frac{\alpha}{2}) \end{aligned} \tag{2.19}$$

where r is the current sonar range reading and is equal to $r = \sqrt{x^2 + y^2}$, while α is the beam angle of the sonar.

Measurement Grouping

A scan frame is labelled as A for the reference and B for the current scan. The composition operator is used to represent the sonar readings with respect to a common reference frame. Let $u_b^a = [u_x, u_y, u_\theta]$ represent the translation and rotation of the reference frame b with respect to the frame a and Q_b^a the associated covariance matrix. Similarly, let x_1^s and x_2^s represent reference and current scans. The transformation of the current to the reference frame is given by

$$x_2^1 = f(x_1^s, x_2^s) = \begin{bmatrix} u_x + x_2\cos(u_\theta) - y_2\sin(u_\theta) \\ u_y + x_2\sin(u_\theta) + y_2\cos(u_\theta) \\ u_\theta + \theta_2 \end{bmatrix} \tag{2.20}$$

Linearising the system around the current estimate and using the first order Taylor approximation, the covariance can be expressed as follows

$$P_2^1 = J_{\oplus 1} P_1^s J_{\oplus 1}^T + J_{\oplus 2} P_2^s J_{\oplus 2}^T \tag{2.21}$$

where

$$J_{\oplus k} = \frac{\partial f(x_1^s, x_2^s)}{\partial (x_k, y_k, \theta_k)} \tag{2.22}$$

Probabilistic Scan Matching

The goal of a scan matching process is to estimate the relative displacement $u_b^a = [u_x, u_y, u_\theta]$ between the reference scan frame 1 (or A) and the current scan frame 2 (or B). This process is usually performed by means of an iterative process. At each iteration k the algorithm establishes, for each point p_i in S_{new} a correspondence q_j in S_{ref} using the current estimate $u_{curr_k}^{ref}$. Next, the new estimate $u_{curr_{k+1}}^{ref}$ is computed as the one that minimises the error of these correspondences. These two steps are repeated until a convergence is achieved.

A- Nearest neighbour data association

Let $p_i = \mathcal{N}(p_i, P_{p_i})$ and $q_j = \mathcal{N}(q_j, P_{q_j})$ be S_{new} and S_{ref} items respectively. Let $p_i = [p_x, p_y]^T$ and $q_j = [q_x, q_y]^T$. To decide whether p_i is compatible or not with q_j , the mahalanobis distance is used. The squared Mahalanobis distance between p_i and q_j is defined as follows:

$$D^2(p_i, q_j) = h_{i,j}^T C_{i,j}^{-1} h_{i,j} \tag{2.23}$$

where $h_{i,j} = h(u_2^1, p_i, q_j)$ computes the difference between p_i and q_j . To calculate this difference, p_i has to be transformed to the reference frame 1 (i.e. A).

$$h_{i,j} = \begin{bmatrix} u_x + p_x \cos u_\theta - p_y \sin u_\theta \\ u_y + p_x \sin u_\theta + p_y \cos u_\theta \end{bmatrix} - \begin{bmatrix} q_x \\ q_y \end{bmatrix} \tag{2.24}$$

Linearising h using the First order Taylor's approximation, the covariance matrix $C_{i,j}$ can be computed as follows

$$C_{i,j} = J_{3_{i,j}} Q_2^1 J_{3_{i,j}}^T + J_{4_{i,j}} P_{p_i} J_{4_{i,j}}^T + P_{q_j} \tag{2.25}$$

where

$$\begin{aligned} J_{3_{i,j}} &= \frac{\partial h(u, pq)}{\partial u} \\ J_{4_{i,j}} &= \frac{\partial h(u, pq)}{\partial p} \end{aligned} \tag{2.26}$$

The matrix Q_2^1 is the covariance associated to u_2^1 . The computation of this matrix will be discussed in section C. Thus, p_i and q_j are compatible if and only if $D^2(p_i, q_j) < \chi_{2,p}^2$, where p is the desired confidence level. For each p_i , the set of

compatible points in S_{ref} is built. Among them, the corresponding point q_j is selected as the one which is closer to p_i in the Mahalanobis sense.

B- Minimisation

The second step in the scan matching process is to find the relative displacement $u_2^1 = [u_x, u_y, u_\theta]$ that minimises the error between pairs of corresponding points. The notation $C = [< a_1, b_1 >, < a_1, b_1 >, \ldots, < a_n, b_n >]$ will be used to denote correspondences between the a_i in S_{ref} and the b_i in S_{curr}. In this work, a Least Square method has been used to minimise the sum of squared Mahalanobis distance between the points in C. The criteria is to minimise the following:

$$\min_u \sum_{i=1}^{n} h_i^T C_{i,i}^{-1} h_i \tag{2.27}$$

where $h_i = h(u, b_i, a_i)$ as defined previously in section A. Linearising h using the first order Taylor approximation, the above equation can be written as follows:

$$\min_u (Ju - A) R^{-1} (Ju - A) \tag{2.28}$$

where

$$J = \begin{bmatrix} J_{3_{1,1}} \\ J_{3_{2,2}} \\ . \\ . \\ . \\ J_{3_{n,n}} \end{bmatrix} \quad A = \begin{bmatrix} J_3 1, 1u - h(u, b_1, a_1) \\ J_3 2, 2u - h(u, b_2, a_2) \\ . \\ . \\ . \\ J_3 n, nu - h(u, b_n, a_n) \end{bmatrix} \tag{2.29}$$

and R is a block diagonal matrix containing the $C_{i,i}$. By using the orthogonality principle, the u that minimises the previous equation is

$$u_{min} = (J^T R^{-1} J)^{-1} J^T R^{-1} A \tag{2.30}$$

The u_{min} is used in the next iteration as the $u_2^1 = [u_x, u_y, u_\theta]$ in order to find correspondence and minimise again. Note that in the first iteration, $u_2^1 = [u_x, u_y, u_\theta]$ is obtained from odometry.

C- Error estimation

The scan matching output is represented as a Gaussian distribution of the form $u_2^1 = \mathcal{N}(u_2^1, Q_2^1)$. This section describes a method to compute the covariance matrix Q_2^1. Let the function $F(u)$ be defined as follows

$$F(u) = \begin{bmatrix} h(u, b_1, a_1) \\ h(u, b_2, a_2) \\ . \\ . \\ . \\ h(u, b_n, a_n) \end{bmatrix} \tag{2.31}$$

The covariance $C_{i,i}$ is known for each $h(u, b_i, a_i)$. Thus, the block diagonal matrix R containing the $C_{i,i}$ represents the covariance of $F(u)$. Linearising $F(u)$ around u and using the first order Taylor approximation, R can be written as follows

$$R = J_5 Q_2^1 J_5^T \tag{2.32}$$

where

$$J_5 = \frac{\partial F(u)}{\partial u} \tag{2.33}$$

Thus, the scan matching covariance Q_2^1 can be computed as follows

$$Q_2^1 = J_5^+ Q (J_5^T)^+ \tag{2.34}$$

where + represents the Moore-Penrose pseudo inverse of a matrix.

2.5.2 Iterative Closest Point with Least Median Error Minimisation (ICP-LMS)

ICP-LMS is based on ICP scan matching algorithm. The important difference is that ICP-LMS minimises the square of median error between corresponding points. The implementation of ICP-LMS is carried out as follows:

Let S_{ref} and S_{curr} represent the current and reference scans, and motion, between the current and reference scan, T_{LMS} initially set to the identical transformation $T_0 = I$ and updated to the motion that is evaluated *best ever* in each trial. At the end of all trials, T_{LMS} satisfies the LMS condition. The overall ICP-LMS scan matching is, therefore, carried out as follows:

- Initialisation: $T_{LMS} = T_0$.
- For n from 1 to N_T, where N_T signifies the number of trials
  - A set of N_S points Q_{RS} is extracted from S_{ref} at random: $Q_{RS} = RS(S_{ref}, N_S)$.

- The Point set Q_{RS} is used by the ICP algorithm with the S_curr to estimate the motion parameters: $T_{ICP} = ICP(Q_{RS}, T_{LMS}, S_{curr})$.
- The estimated motion is evaluated by $MS(S_{ref}, T_{ICP}, S_{curr})$.
- If $MS(S_{ref}, T_{ICP}, S_{curr}) < MS(S_{ref}, T_{LMS}, S_{curr})$, then $T_{LMS} = T_{ICP}$

- The resultant motion T_{LMS} satisfies:
  $MS(S_{ref}, T_{LMS}, S_{curr}) = \min_{1<n<N} MS(S_{ref}, T_{LMS}, S_{curr})$ and this is the result of LMS scan matching (motion estimation) between S_{ref} and S_curr.

Where
N_T is the number of trials;
N_S is the number of random samples;
Q_{RS} is an array with randomly extracted scan points from S_{ref};
RS is a function to do the random sampling;
T_{LMS} is the motion estimated through Least Median;
T_{ICP} is motion estimated through ICP;
MS is used to compute the squared mean error between corresponding points, generated using the given transformation, i.e. T_{LMS} or T_{ICP}.

2.5.3 Numerical Results

The method of psIC is implemented in the Matlab simulation environment. The odometry and sonar sensor error are taken into account in the implementation. This method is analogous to ICP and IDC, but it groups sonar readings prior to the scan matching and corrects the trajectory after the scan matching. This method works for environment which are non-structured. No assumption in the geometry of the environment is taken into account. The environment considered for the simulated tests is a generic non-structured 3D structure, with the AUV moving towards the structure.

Figure 2.9 shows the vehicle trajectory (starting from the left, going towards the structure) and the estimated trajectory of the vehicle, which is a bit diverted from the ground truth. Scan matching using psIC gives a relatively poor result in estimating the vehicle state. This is due to two sources of errors: the sonar data and the motion model. In addition, this method is shown to constantly diverge from the ground truth. This is because there is no error correction method like the update model of EKF.

Figure 2.10 shows the plots of errors and uncertainties of the psIC in the estimation of the vehicle state. The current scan is transformed to the previous scan frame using the motion from odometry. The matching is performed on the previous reference system. The estimated trajectory is improved and the error level is reduced from 6.0

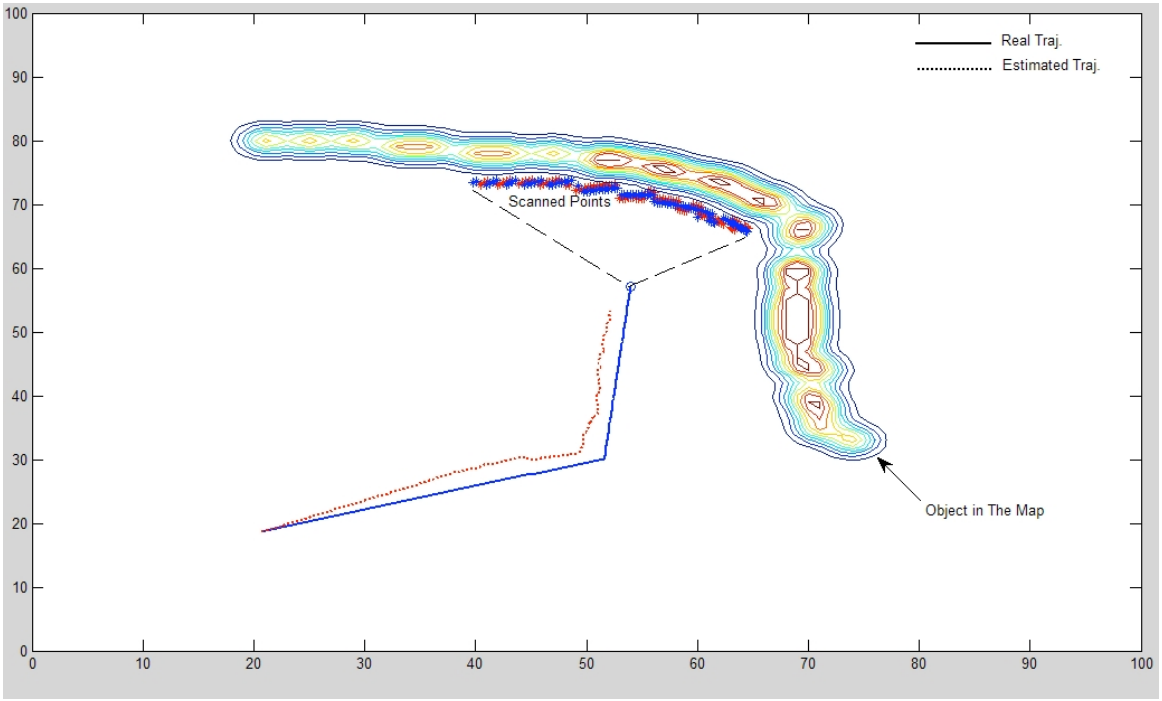


Figure 2.9: psIC Real and Estimated Localisation Trajectories

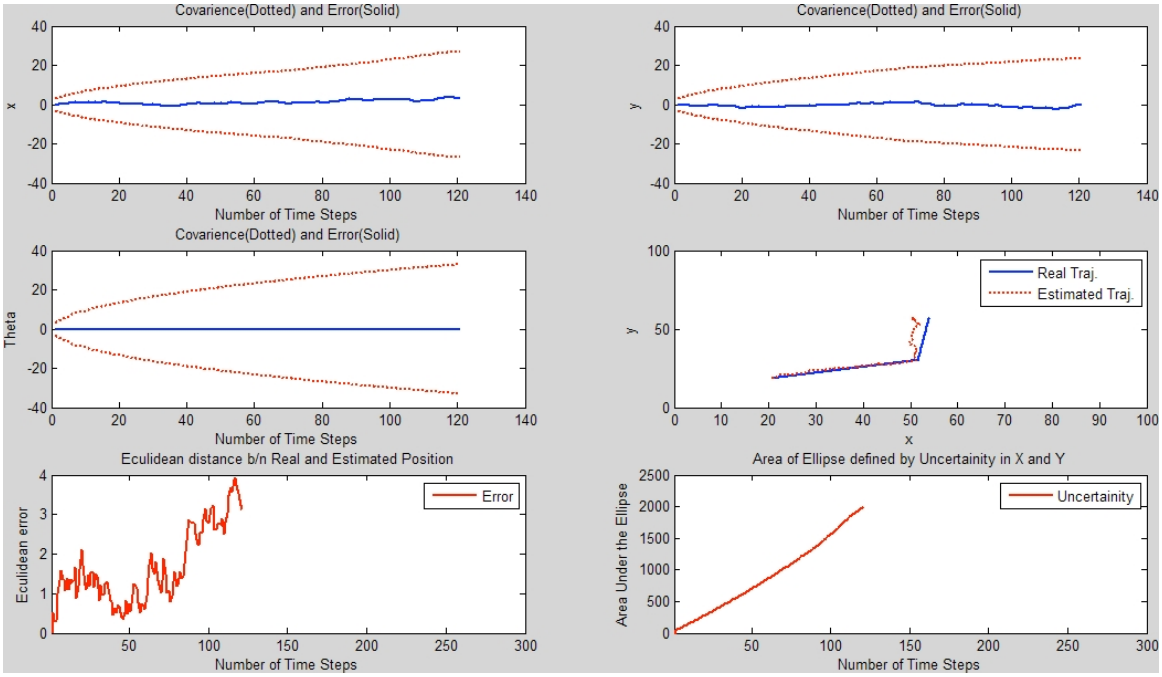


Figure 2.10: The Error and Uncertainty of the psIC Scan Matching

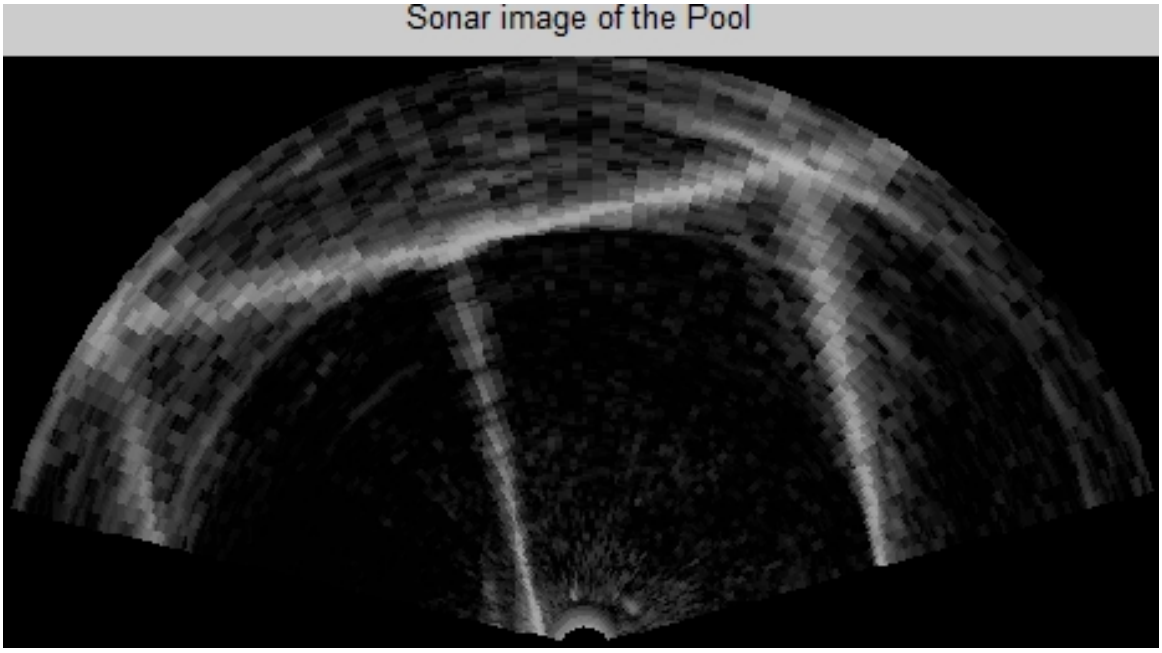


Figure 2.11: Sonar Image taken at the left corner of the pool

m (motion model) to 2.0 m (Scan matching). In order to reduce the computational load, a smaller field of view could be used. However this means a greater error in localisation and in limited information about the environment.

2.5.4 Experimental Results

The Scan matching using ICP-LMS is tested using the Cartesian Robot described in section 1.6.1. A Tritech mechanically scanned profiling sensor is mounted on the Robot. The pool is 4x3 meters and 2 meters deep. The localisation is performed on a horizontal plane at about 1 meter altitude. In this section, the scan matching is performed without assuming a priori knowledge of the motion. Scan matching is performed using the sonar data only.

Sonar images are processed to extract features. Figure 2.11 shows a sample sonar scan image in the environment of interest.

In order to remove the outliers (as in Figure 2.12), range distance of each point is compared with its left and right adjacent points. If the distance between these points exceeds a threshold of 0.25 meters the point is regarded as an outlier. Figure 2.13 shows the result of the re-enforced feature extraction.

The results are also compared with the standard ICP using Least Mean of Squares error. It is shown that ICP-LMS offers better estimate and a smaller error compared to ICP. But this advantage comes at the expense of an increased computational complexity.

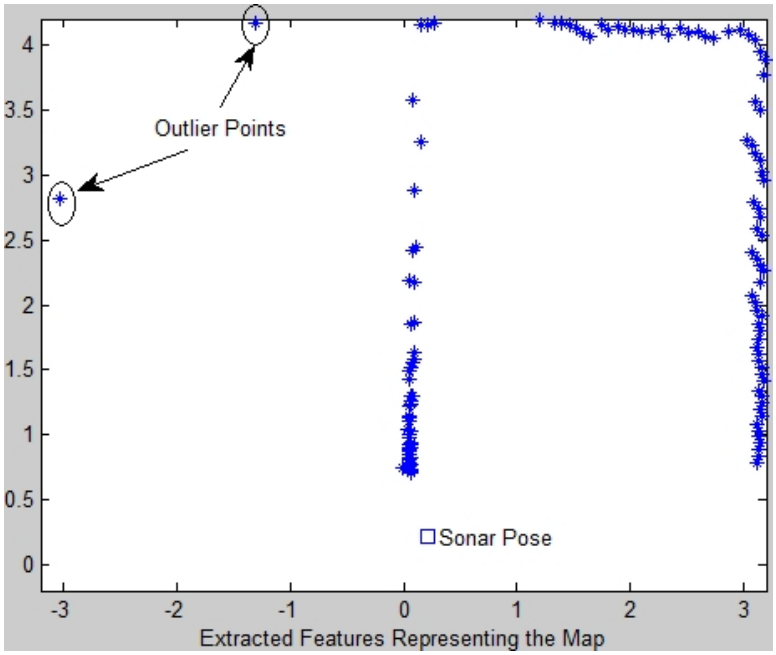


Figure 2.12: The processed image representing the walls of the pool. The small rectangle refers the sonar pose. Two outlier points are extracted. This is due to the reflection effect on the sonar image.

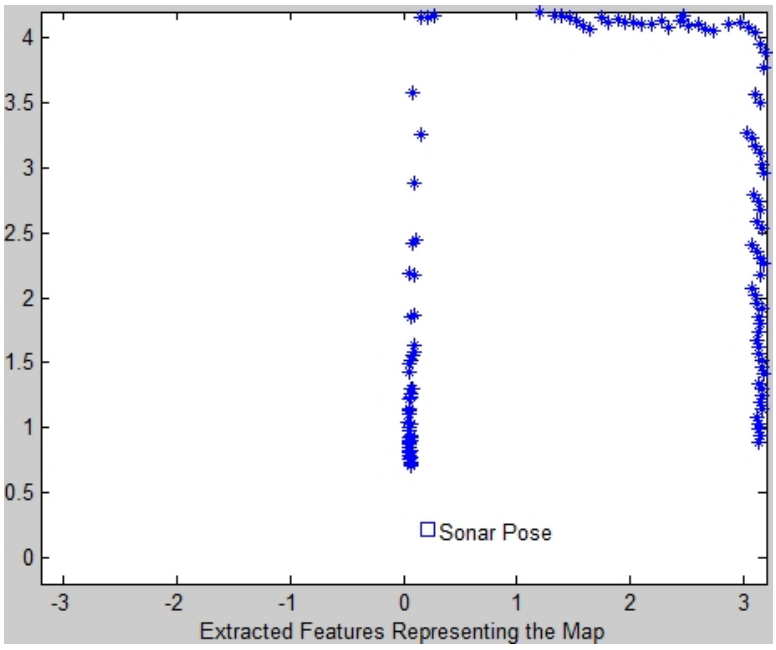


Figure 2.13: The processed image representing the walls of the loop. The small rectangle refers the sonar pose. The outliers are removed using an improved feature extraction.

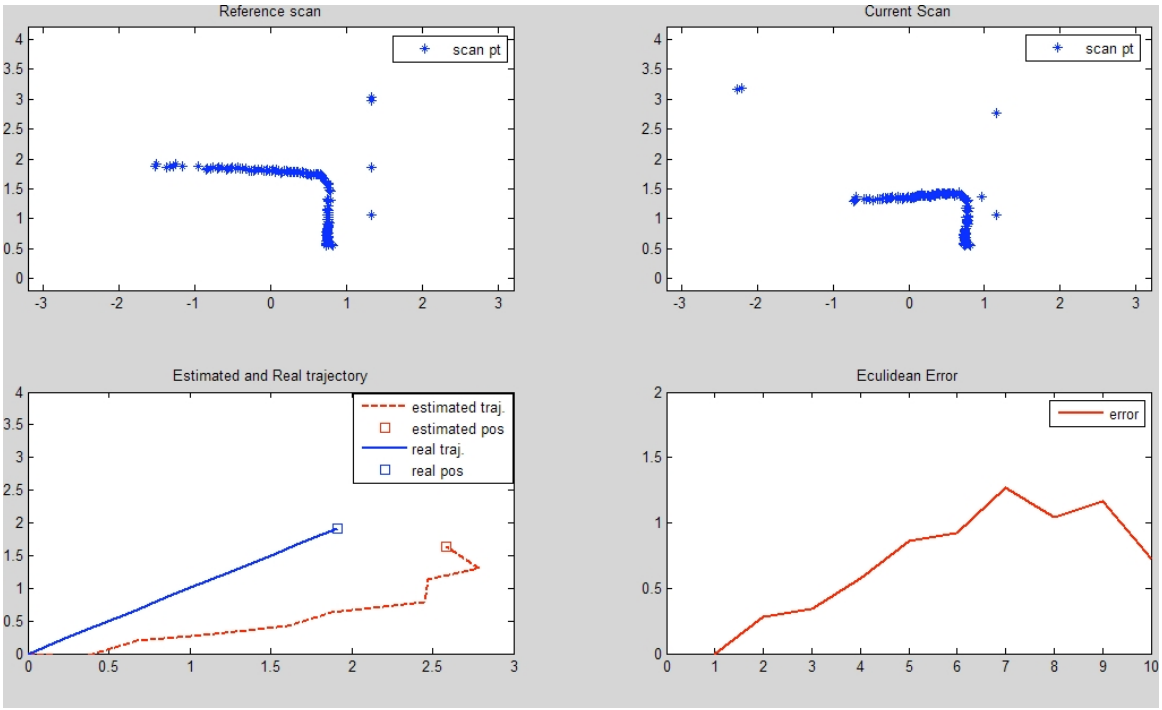


Figure 2.14: ICP localisation without using a motion model

The ICP starting with an identity transformation (i.e. with no motion model) is shown to give a growing error as the vehicle keeps navigating, as in Figure 2.14. The error increases up to 1.5 meters. This is due to the fact that ICP is easily affected by little number of outlier scan points.

Figure 2.15 shows the results of the implementation of ICP-LMS method of localisation. In a similar approach like the normal ICP, no initial motion estimate is assumed and the matching is performed using the sequence of scanned images. As expected, the error grows over time. However, the results of ICP-LMS are much better than standard ICP.

Varying the sonar range, the results of the scan matching can change significantly. Figure 2.16 shows the error with a range varying from 3 to 7 meters. In general the error increases when the range increases. This is because more outliers are considered for bigger ranges, and they then affect the overall performances.

2.6 Set membership methods

2.6.1 Introduction

This section presents a less-known approach to localisation, using a Set-membership approach. Set-membership methods have often been considered for the localisation of robots [75][40], in the case where the problem is linear and also [14] when the robot is

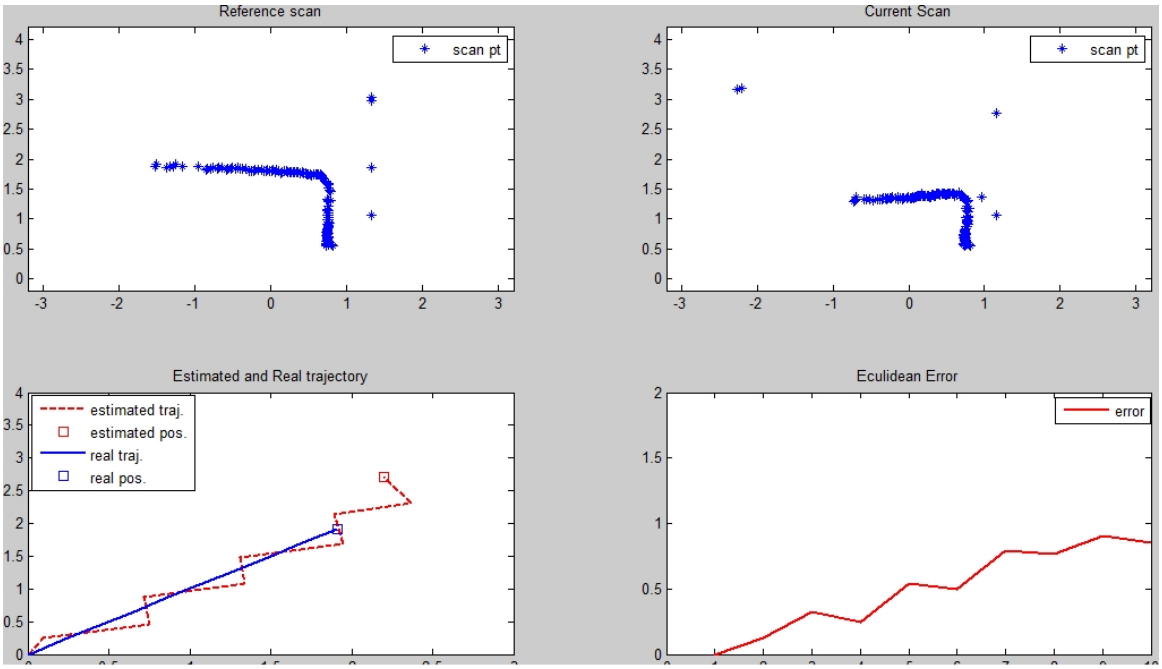


Figure 2.15: ICP-LMS localisation without using a motion model

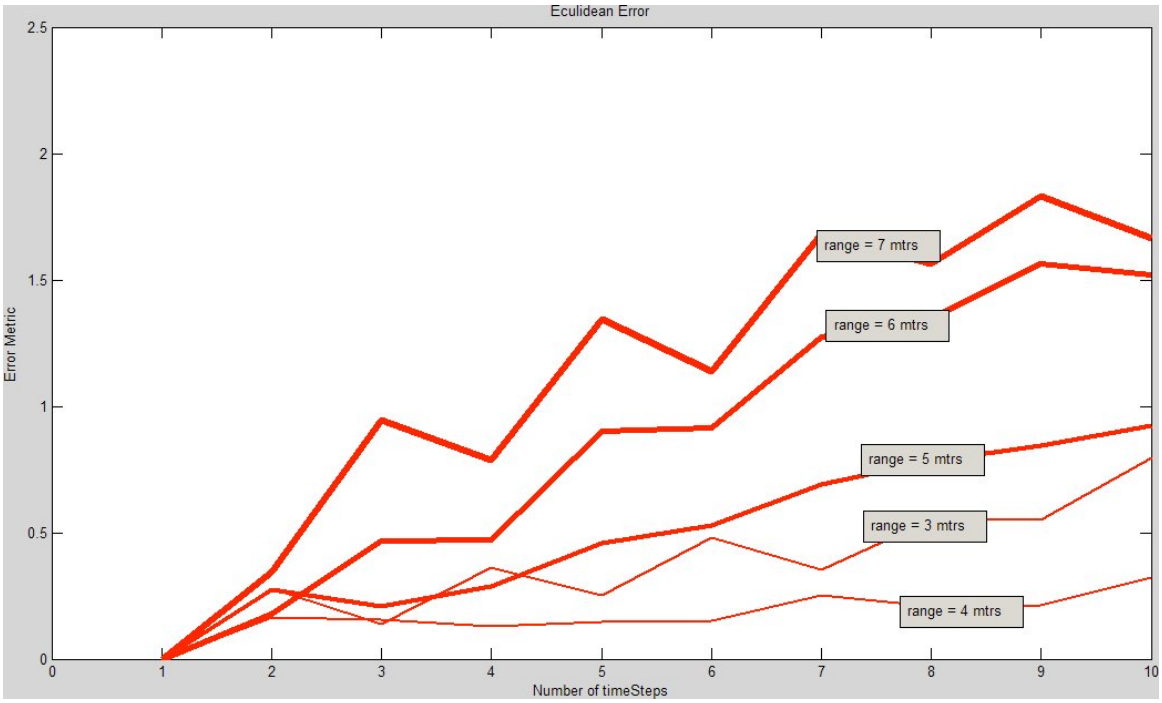


Figure 2.16: Euclidean Error for ICP-LMS assuming a sonar range dimension of 3,4,5,6 and 7 meters.

underwater. In situations where strong non-linearities are involved, interval analysis has been shown to be useful (see, e.g., [74], where the first localisation of an actual robot has been solved with interval methods). Another strong point of set membership methods is the ability to deal with outliers (see [45]). In the Set Membership formulation, both input data and computed robot position are represented by their respective belonging sets. Constraints between the position of the robot and the sensor observations are used to contract the actual position set, *i.e.* reducing its size thus increasing the estimates precision. In this section a presentation of the technique is given and experimental results are presented. A critical analysis compared with the Particle Filter technique previously presented is then given. The work presented in this section was performed with Jan Sliwka, [100].

2.6.2 Definitions and notations

Interval : An interval is a connected and closed subset of $\mathbb{R}$. If x is a real variable, $[x]$ is the interval containing this variable. $[x]$ is called the domain of x. An interval has an upper and lower bound which is noted as follows $[x] = [x^-, x^+]$. $\mathbb{IR}$ is the set of all the real intervals. $\mathbb{IN}$ is the set of natural number intervals. $w([x]) = x^+ - x^-$ is called width of $[x]$. For example, $\emptyset, \{-1\}, [-1, 1], [-1, \infty], \mathbb{R}$ are intervals.

Box : A box of $\mathbb{R}^n$ is defined by a Cartesian product of intervals. A box can be also considered as an interval vector. If $\mathbf{x} = (x_1, .., x_n) \in \mathbb{R}^n$ is a real variable vector we denote by $[\mathbf{x}] = ([x_1], .., [x_n])$ the box containing this variable. For example, $[1, 3] \times [2, 4]$ is a box of $\mathbb{R}^2$.

CSP : A constraint satisfaction problem (or CSP) is defined by a set of constraints $C_1, .., C_n$, a vector of variables $\mathbf{x} = (x_1, .., x_m)$ and the domain $\mathbf{D} = D_1 \times .. \times D_m$ of possible values of $\mathbf{x}$. Only continuous CSP are considered, where the domain $\mathbf{D}$ is a subset of $\mathbb{R}^m$. The constraints are linear or nonlinear equations or inequalities

$$\begin{array}{l} g_i : \mathbb{R}^m \to \mathbb{R}, h_i : \mathbb{R}^m \to \mathbb{R} \\ C_i : g_i(\mathbf{x}) \leq 0, i \in \{1, .., k\} \\ C_i : h_i(\mathbf{x}) = 0, i \in \{k+1, .., p\}. \\ \mathbf{x} \in \mathbf{D} \end{array} \tag{2.35}$$

A more general notation can be used to represent such a CSP

$$\begin{array}{l} \mathbf{f}_i : \mathbb{R}^m \to \mathbb{R}^\ell \\ C_i : \mathbf{f}_i(\mathbf{x}) \in [\mathbf{y}_i], \\ \mathbf{x} \in \mathbf{D}, [\mathbf{y}_i] \subset \mathbb{R}^\ell, i \in \{1, .., n\} \end{array} \tag{2.36}$$

2.6.3 Mathematical formulation

A dynamic system such as an underwater robot can usually be characterised by discrete-time dynamic equations

$$\begin{aligned} &\mathbf{f}_k : \mathbb{R}^m \to \mathbb{R}^m, \mathbf{g}_k : \mathbb{R}^m \to \mathbb{R}^\ell \\ &\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) \\ &\mathbf{y}_k = \mathbf{g}_k(\mathbf{x}_k). \end{aligned} \tag{2.37}$$

where $\mathbf{x}_k$ is the state of the system, $\mathbf{y}_k$ is the output vector, $\mathbf{f}_k$ is the evolution function and $\mathbf{g}_k$ the observation function. The input of the system is enclosed in the expression of $\mathbf{f}_k$. The noise (due to model imperfection) is neglected. The noise can be actually added as a state of the system but this work focuses on the position to simplify the different formulations. In our case, $\mathbf{x}_k$ is the robot pose, $\mathbf{f}_k$ characterises robots dynamics, $\mathbf{y}_k$ is the measurement vector (here sonar distance to obstacle measurements). $\mathbf{y}_k$ and $\mathbf{x}_k$ are related by the observation function $\mathbf{g}_k$ which express in our case geometrical relations between the position, the measurements and the map. Denote by $[\mathbf{y}_k] \subset \mathbb{R}^\ell$ the domain of the measurement $\mathbf{y}_k$. Denote by $[\mathbf{x}_{k,0}] \subset \mathbb{R}^m$ the prior domain of $\mathbf{x}_k$. Using state equation in (2.37), the problem of estimation of $\mathbf{x}_k$ can be cast into the following set of equations

$$\begin{aligned} &\mathbf{g}_k(\mathbf{x}_k) = \mathbf{y}_k \\ &\mathbf{g}_{k-1} \circ \mathbf{f}_{k-1}^{-1}(\mathbf{x}_k) = \mathbf{y}_{k-1} \\ &... \\ &\mathbf{g}_{k-n+1} \circ \mathbf{f}_{k-n+1}^{-1} \circ ... \circ \mathbf{f}_{k-1}^{-1}(\mathbf{x}_k) = \mathbf{y}_{k-n+1} \\ &\mathbf{x}_k \in [\mathbf{x}_{k,0}], \mathbf{y}_i \in [\mathbf{y}_i], i \in \{k-n+1, .., k\}, \end{aligned} \tag{2.38}$$

this set of equations can also be called a continuous *constraint satisfaction problem* or *CSP*. The CSP can be solved using set membership methods as shown in section 2.6.5. It is possible to assume that at the time step k the state $\mathbf{x}_k$ (position of the robot) is *a priori* completely unknown. In that case, the prior domain of membership of $\mathbf{x}_k$ is $[\mathbf{x}_{k,0}] = \mathbb{R}^m$. In the localisation jargon, this corresponds to the *global localisation*.

On the other hand, it is possible to exploit the fact that the current state of the system $\mathbf{x}_k$ depends of its previous state $\mathbf{x}_{k-1}$ since

$$\begin{aligned} &\mathbf{f}_k : \mathbb{R}^m \to \mathbb{R}^m \\ &\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}). \end{aligned} \tag{2.39}$$

In the localisation jargon, this corresponds to the *dynamic localisation* or position tracking. The prior domain of membership of the state $\mathbf{x}_k$ depends on the domain of $\mathbf{x}_{k-1}$ which is supposed to be computed. Denote by $\mathbb{X}_{k-1}$ the solution set of the CSP

corresponding to the problem of estimation of $\mathbf{x}_{k-1}$ using set membership methods. We have

$$[\mathbf{x}_{k,0}] = [\mathbf{f}_{k-1}(\mathbb{X}_{k-1})] \tag{2.40}$$

where $[\mathbf{f}_{k-1}(\mathbb{X}_{k-1})]$ is the smallest box enclosing the set $\mathbf{f}_{k-1}(\mathbb{X}_{k-1})$. The advantage of dynamic localisation *versus* global localisation is to reduce the size of the search space and as a consequence reduce the computation time for the same precision requirements. In order to simplify the theoretical explanations the following notations are considered:

$$\begin{array}{l} \forall i \in \{1,..,n\}, \mathbf{h}_i : \mathbb{R}^m \to \mathbb{R}^\ell, \mathbf{z}_i \in \mathbb{R}^\ell, \mathbf{x} \in \mathbb{R}^m, [\mathbf{x}_0] \subset \mathbb{R}^m, \\ \mathbf{h}_1 = \mathbf{g}_k \\ \mathbf{h}_2 = \mathbf{g}_{k-1} \circ \mathbf{f}_{k-1}^{-1} \\ \mathbf{h}_i = \mathbf{g}_{k-i+1} \circ \mathbf{f}_{k-i+1}^{-1} \circ ... \circ \mathbf{f}_{k-1}^{-1}, i \in \{3,..,n\}, \\ \mathbf{z}_i = \mathbf{y}_{k-i+1}, i \in \{1,..,n\} \\ \mathbf{x} = \mathbf{x}_k, [\mathbf{x}_0] = [\mathbf{x}_{k,0}], \end{array} \tag{2.41}$$

The CSP becomes

$$\begin{array}{l} \mathbf{h}_1(\mathbf{x}_k) = \mathbf{z}_1 \\ ... \\ \mathbf{h}_n(\mathbf{x}_k) = \mathbf{z}_n \\ \mathbf{x} \in [\mathbf{x}_0], \mathbf{z}_i \in [\mathbf{z}_i], i \in \{1,..,n\}. \end{array} \tag{2.42}$$

2.6.4 Relaxed resolution of the system of equations

Consider the CSP defined in (2.42). The solution set $\mathbb{S}$ of such CSP is

$$\mathbb{S} = \{\mathbf{x} \in [\mathbf{x}_0], \forall i \in [1..n], \mathbf{h}_i(\mathbf{x}) = \mathbf{z}_i, \mathbf{z}_i \in [\mathbf{z}_i], [\mathbf{z}_i] \subset \mathbb{R}\} \tag{2.43}$$

or in a more compact form

$$\mathbb{S} = \{\mathbf{x} \in [\mathbf{x}_0], \forall i \in [1..n], \mathbf{h}_i(\mathbf{x}) \in [\mathbf{z}_i], [\mathbf{z}_i] \subset \mathbb{R}\} \tag{2.44}$$

In some cases, this CSP doesn't admit any solution. In the context of localisation, that will be the case when some of the measurements are erroneous *i.e.* there are outliers. Since the equations come from measurements, the equations coming from erroneous measurements are *a priori* not satisfied. Dealing with outliers has already been considered by several authors, in a set membership context (see, *e.g.*, [83], [63], [91], [60] ,[45]). In this case, we define a solution set $\mathbb{S}_q$ where $\mathbf{x} \in \mathbb{S}_q$ satisfies only a part of equations from the set of equation. This problem is called a relaxed CSP.

A q-relaxed resolution of the CSP (2.42) is searching for a solution set $\mathbb{S}_q$ where $\mathbf{x} \in \mathbb{S}_q$ satisfies at least $n-q$ among n equations *i.e.* searching for the following solution set

$$\mathbb{S}_q = \{\mathbf{x} \in \mathbb{R}^m, \exists \mathbb{I} \subset \{1,..,n\}, Card(\mathbb{I}) = n-q, \forall i \in \mathbb{I}, \mathbf{h}_i(\mathbf{x}) \in [\mathbf{z}_i], [\mathbf{z}_i] \subset \mathbb{R}\}. \quad (2.45)$$

2.6.5 Using set membership methods

Set membership methods allow to manipulate sets (see [79] and [46]). As an example, considering $\mathbb{A}, \mathbb{B}, \mathbb{C}$ subsets of $\mathbb{R}^n$ and a function f, those methods allow to compute intersection $\mathbb{A} = \mathbb{B} \cap \mathbb{C}$, union $\mathbb{A} = \mathbb{B} \cup \mathbb{C}$, set inversion $\mathbb{A} = f^{-1}(\mathbb{B})$, image of a set by a function $\mathbb{A} = f(\mathbb{B})$. It is possible to implement those operations on a computer in a form of solvers such as QUIMPER [19]. The solution of the CSP in (2.42) can also be characterised by an intersection of sets. Denoting by $\mathbb{X}_i$ the set of points $\mathbf{x}$ which satisfy the i^{th} constraint of the CSP (defined in 2.42) $\mathbf{h}_i(\mathbf{x}) \in [\mathbf{z}_i]$, the formulation becomes:

$$\mathbb{X}_i = \{\mathbf{x} \in \mathbb{R}^m, \mathbf{h}_i(\mathbf{x}) \in [\mathbf{z}_i], [\mathbf{z}_i] \subset \mathbb{R}\} = \mathbf{h}_i^{-1}([\mathbf{z}_i]), i \in \{1,..,n\}. \quad (2.46)$$

with

$$\mathbb{S} = \bigcap_{i \in \{1..n\}} \mathbb{X}_i. \quad (2.47)$$

In case of outliers, since not all the equations are satisfied, there is no point which satisfy all the equations thus $\mathbb{S}$ is an empty set. The idea which have been introduced in [45] is to define $\mathbb{S}_q$ as a special intersection of the $\mathbb{X}_i$ sets defined in (2.46) called the q-relaxed intersection where the number q corresponds to the number of outliers in the data. Suppose the i^{th} measurement is an outlier. The corresponding set $\mathbb{X}_i$ doesn't necessarily contain any viable solution. As such, the set $\mathbb{X}_i$ shouldn't be taken into consideration during the intersection process hence the relaxed intersection.

The q-relaxed intersection (see [45]) of.n sets $\mathbb{X}_1,..,\mathbb{X}_n$ also denoted by $\bigcap_{i \in \{0..n\}}^{\{q\}} \mathbb{X}_n$ is a set of points which are in at least $n-q$ sets among the $\mathbb{X}_i, i \in \{1,..,n\}$ sets *i.e.*

$$\bigcap_{i \in \{1,..,n\}}^{\{q\}} \mathbb{X}_i = \{\mathbf{x} \in \mathbb{R}^m, \exists \mathbb{I} \subset \{1,..,n\}, card(\mathbb{I}) = n-q, \forall i \in \mathbb{I}, \mathbf{x} \in \mathbb{X}_i\} \quad (2.48)$$

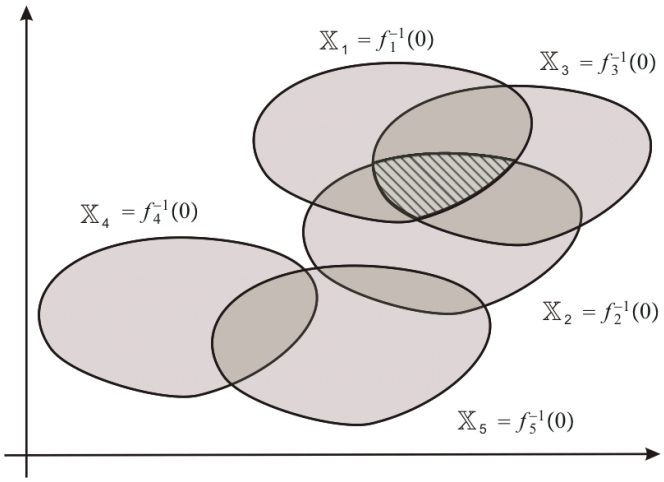


Figure 2.17: Illustration of the relaxed intersection of 5 sets $\mathbb{X}_1, \mathbb{X}_2, \mathbb{X}_3, \mathbb{X}_4$ and $\mathbb{X}_5$. The hatched set corresponds to the 2-relaxed intersection of those sets.

Figure 2.17 shows the q-relaxed intersection of sets $\mathbb{X}_1, \mathbb{X}_2, \mathbb{X}_3, \mathbb{X}_4$ and $\mathbb{X}_5$. We have

$$\bigcap_{i\in\{1..5\}}^{\{0\}} \mathbb{X}_i = \bigcap_{i\in\{1..5\}} \mathbb{X}_i = \emptyset$$

$$\bigcap_{i\in\{1..5\}}^{\{2\}} \mathbb{X}_i = \mathbb{X}_1 \cap \mathbb{X}_2 \cap \mathbb{X}_3 \text{ (hatched set in Figure 2.17)}$$

$$\bigcap_{i\in\{1..5\}}^{\{5\}} \mathbb{X}_i = \bigcup_{i\in\{1..5\}} \mathbb{X}_i.$$

The number of outliers q is usually unknown but quite often the number of those outliers is bounded and never go beyond a maximum number $q_{\max}$. The solution of the relaxed CSP becomes

$$\mathbb{S}_{q_{\max}} = \bigcap_{i\in\{1..n\}}^{\{q_{\max}\}} \mathbb{X}_i, \tag{2.49}$$

this solution is guaranteed as long as the assumption about the number of outliers ($q < q_{max}$) is respected.

The dynamic function

The robot evolution can be characterised by the following differential equation

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} v * \cos(\theta) \\ v * \sin(\theta) \\ \omega \\ a \\ a_\omega \end{pmatrix} \tag{2.50}$$

where (x, y, θ) is the pose of the robot, v its speed, ω its rotation speed, a is the acceleration and a_ω is the rotation acceleration..Using Euler discretisation we obtain

$$\begin{aligned} x_{k+1} &= x_k + v_k \cos(\theta_k) dt \\ y_{k+1} &= y_k + v_k \sin(\theta_k) dt \\ \theta_{k+1} &= \theta_k + \omega_k dt \\ v_{k+1} &= v_k + a_k dt \\ \omega_{k+1} &= \omega_k + a_{\omega,k} dt \end{aligned} \tag{2.51}$$

where (x_k, y_k, θ_k) is the state of the robot, v_k is its speed, θ_k is its orientation and ω_k is the rotation speed at time step k. The acceleration a_k and rotation acceleration $a_{\omega,k}$ are considered as input noise.

2.6.6 Solving

Once the expression of the relaxed CSP is found, the RSIVIA solver (see [45]) can be used to compute the solution set.

Results

Figure 2.18 shows the reconstructed trajectory comparing it to the GPS reference trajectory (in black). The algorithm will start by performing a global localisation in order to find the initial position. In this case, the searching area is the whole marina. This step takes several seconds since the required precision is big with respect to the size of the searching area. Once the position has converged, the algorithm switches to position tracking. The algorithm computes the trajectory in the form of a set of boxes. The real trajectory (which is a set of points) is included in this set of boxes. Usually the trajectory formed by the centres of those boxes can be considered as an approximation of the real trajectory.

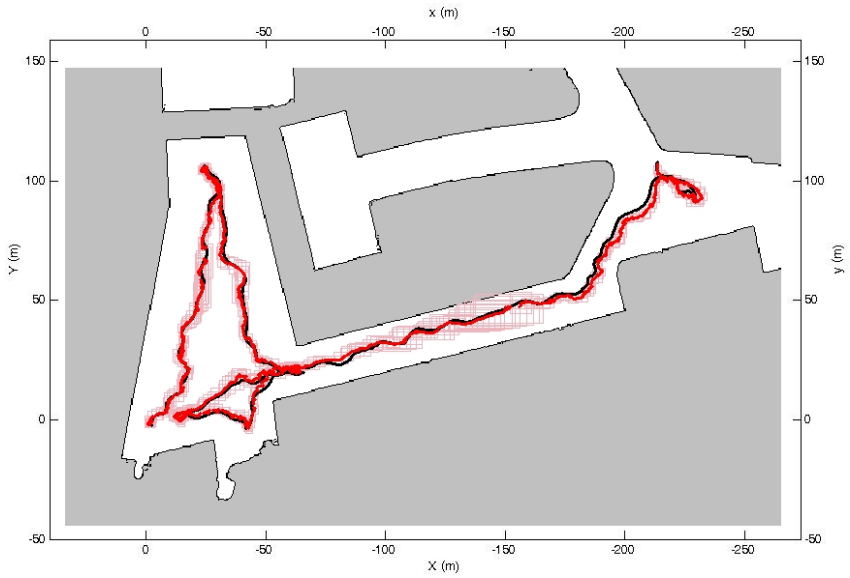


Figure 2.18: Comparing the GPS trajectory (in black) with the trajectory computed using set membership methods.

Advantages and drawbacks

The main advantages of this method are guaranteed results and robustness to outliers which varies in our case from 0 to 15%. Another advantage is that the algorithm is capable of detecting inconsistent situations such as an error in the map model or the sensor malfunction. Another advantage not displayed in this application is the possibility to use non-linear equations for both evolution and observation functions. The main disadvantage of this method is that often the output is a large solution sets for positions, without knowing where is the biggest probability of occurrence. The center of the set is usually taken as a better approximation.

2.7 Conclusions

This Chapter has reviewed the main techniques currently used for localisation, providing both an analysis of the related work, the mathematical formulation of Bayesian filtering, and some numeric and experimental results using standard techniques such as Kalman Filters, Scan Matching and Set Membership approach. From the analysis of the different approaches, the topics this thesis aims to address in this area are related to a localisation system for underwater vehicles which:

- is not dependent on external aid;
- solves both global localisation and position tracking;

- is able to cope with high level of noisy data and imprecisions in the previous knowledge of the map;
- is able to recover from wrong convergences;
- is robust and reliable not just in simulated tests, but also integrated in the robot architecture and tested over long trajectories in the field.

The next Chapter will present a localisation system with some novelty aspects covering for all these areas.

Chapter 3

Novel approaches for AUV passive localisation

3.1 Introduction

In the previous Chapter several methods for AUV passive localisation were presented. This chapter will now focus on novel approaches which lead to more general, more robust and faster solutions. The proposed localisation system, as outlined at the end of the previous Chapter, needs to address the following items:

- it should not be dependent on external aid;
- it should solve both global localisation and position tracking;
- it should be able to cope with high level of noisy data and imprecisions in the previous knowledge of the map;
- it should be able to recover from wrong convergences;
- it should be robust and reliable not just in simulated tests, but also integrated in the robot architecture and tested over long trajectories in the field.

This Chapter is organised as follow:

- section 3.2 will present a novel localisation approach for an AUV, based on particle filters (PF). Both results from simulations and field trials will be presented and comparison with other techniques will be highlighted;
- section 3.3 will present the case of partially known environment analysing the performances with varying parameters;

- section 3.4 will present a localisation algorithm, based on fusion between two modules, running an EKF and a PF respectively.

Finally, conclusions will summarise the main achievements presented in this Chapter, presenting the need of an *active* approach, which will be the topic for the following Chapters.

3.2 Particle Filters for Localisation in Distinctive Environment

Among the several techniques presented in the last Chapter, the chosen one as a base to develop a more robust and general system is based on particle filters. Particle filter techniques were chosen because they can handle estimation of non Gaussian and non linear processes. This is very important because non-linearities are very frequent in AUVs, both in the motion model specification and in the observation process. Additionally, the noise cannot be modelled as Gaussian in many situations. Another advantage of using particle filters is that it does not require any assumption on the initial position and orientation of the vehicle. In order for a particle filter to work, the sensed environment needs to be distinctive enough. By distinctive the author means that the sensor's measures vary significantly with position. A typical example is the navigation in man-made environments, like marinas, or navigation close to underwater structures, being them either natural or artificial, like an off-shore underwater oil infrastructure.

The standard approach presents two main problems. The first one is the high number of particles required, in order to explore the state space, resulting in an increase of the computational power needed. The other major issue in particle filter approaches is the *sample impoverishment problem*, i.e. the loss of diversity for the particles to adequately represent the solution space [17]. In our approach, both these problems are addressed, as shown next.

3.2.1 Particle Resampling

The standard SIR algorithm for particle resampling lets the particles with high weight reproduce, while the particle with low weight are more unlikely to survive. However, the resulting probability density function (*pdf*) at time t is depending only on the *pdf* at time $t-1$. In time, this means that only a small part of the state space is represented by the particles and the system cannot recover from an incorrect estimation of the vehicle's position (due to sensor noise for instance). In the proposed system, at each step, a portion of the particles is instantiated randomly in the state space. Thus, the resampling algorithm is built with two modules. The first one is a standard SIR module, returning $N-k$ particles. The second module returns k particles, created randomly. The combination of these two modules constitutes the resampling step in the proposed system. The algorithm is then able to recover in case of a wrong convergence, as shown in the experimental results section. The benefits on the computational point of view are also relevant since there is no need to instantiate a high number of particles. Even if in the initial step there are no particles near the real position of the vehicle, the proposed solution is still able to find the correct position, after some time, thanks to the partial random resampling.

3.2.2 Sensor model and likelihood calculation

The sensor used is a forward looking imaging sonar, used as profiler. The XY sonar image, easy to understand for humans is actually often more complex to analyse for a computer. Additionally the transformation is not needed, thus working with $\rho\theta$ sonar images is to be preferred. Figure 3.1 shows a sonar image taken with a Tritech Micron sonar (see 1.3.3), in the OSL water tank (see 1.7.1), with a range of $3m$. In order to arrive to range values, each beam (line) of the $\rho\theta$ sonar image is scanned, and the first high intensity bin (pixel) is highlighted. Knowing the sonar range, it is possible to therefore calculate the range to the closest obstacle. This approach is also useful to discard sonar echo. Among the parameters to tune, there is the level of threshold for high intensity. In most of the cases, it was predefined, though some work on adaptive threshold could be applied. It is important to remove the noise close to the sonar and the vehicle imprint, when this shows up on the image, according to the mounting position. Overall, this method was proven to be reliable and robust to outliers, following many experimental validation.

Figure 3.2 shows the problem of the likelihood calculation for a particle at a different location than the robot. Considering range values only, the likelihood calculation is therefore an assessment on similarities among two arrays of distances. The first one is the array generated by the real sonar system, whilst the second one is simulated,

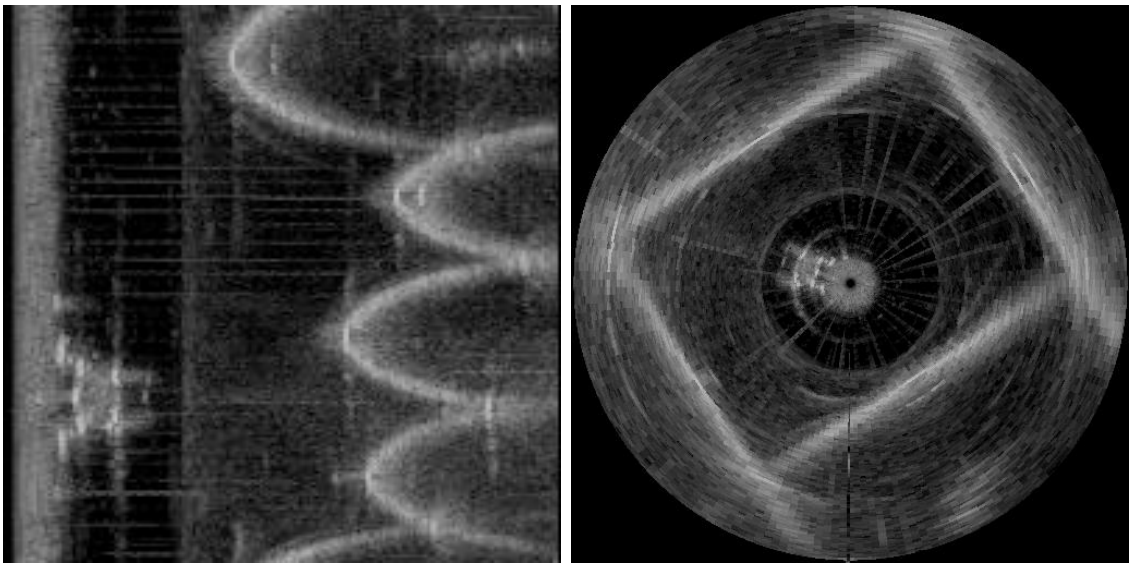


Figure 3.1: $\rho\theta$ sonar image (left) and XY sonar image (right). It is also possible to see the reflection of the *NessieIV* AUV, under which the sonar is mounted.

based on the possible position of the vehicle (given by the particle) and the knowledge of the environment (*a priori* map). For the simulated setup, a raytracing algorithm was used to compute the intersection between the sonar beams and the environment. In order to determine the array of distances for each particle, an alternative solution to raytracing has been used. As the map given by [94] is a set of lines, a geometrical approach, based on line intersection, is much faster than and as accurate as raytracing. For the real tests, sonar data processing is required in order to get the array of distances. The imaging sonar returns for each beam an intensity array. To transform this array in a distance value, a threshold is applied in order to separate the acoustic imprint left by an object in the image, from the noisy background data. Once the two arrays of distances have been computed, the next problem is the likelihood calculation. The likelihood value should reflect the similarity of the two arrays. To let the reader understand better the proposed approach, the case of arrays composed by a single element each (r and s) will be now described. In this case, the likelihood is given by the following equation:

$$L(x) = \frac{1}{\sqrt(2\pi)} \exp -\frac{1}{2}x^2 \tag{3.1}$$

where $x = r - s$, r represents the real value of distance, given by the vehicle, and s represents the simulated value, given by the particles. In the case of arrays with more than one value, the same procedure can be applied to all indexes of the array, producing likelihood array. In order to calculate a single likelihood value, a mean value solution was chosen. The likelihood function calculated in this way is therefore

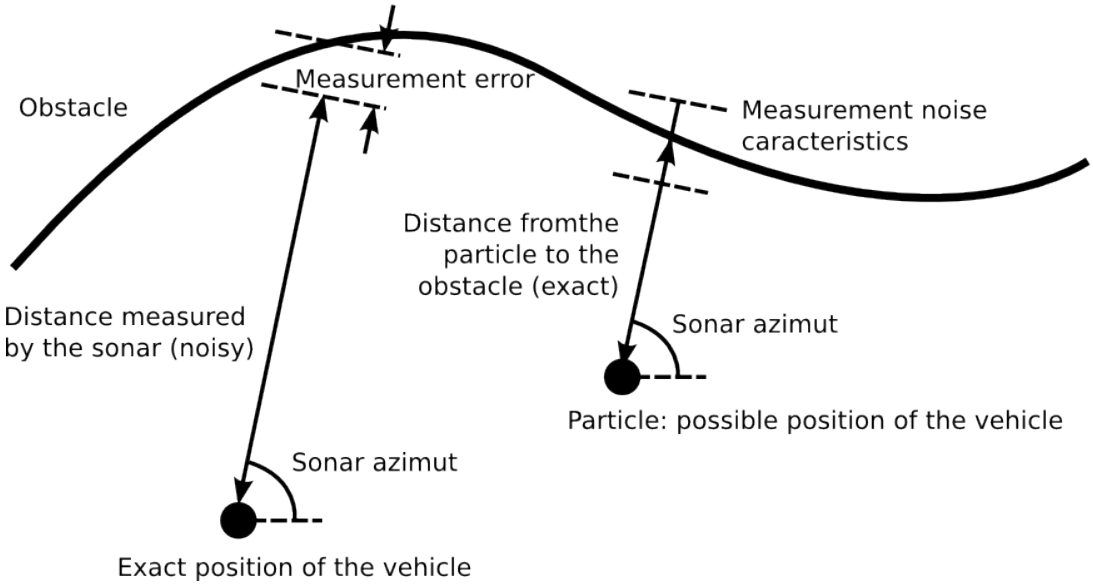


Figure 3.2: Illustration of the calculation of the likelihood for the particle filter.

a mixture of Gaussians. A pure Gaussian likelihood was also tested, multiplying the likelihoods of the single indexes, like a joint probability of independent variables. Both methods were proven valid and reliable. However, the product one is more selective but also more sensitive to noise, as a few bad index likelihoods have a great impact on the overall likelihood of the particle. On the other hand, the average method seems to preserve better the diversity of the particles in the solution state. Thus, it is to be preferred when the particles used are few, while the product is to be preferred when there are many particles in the same area, to discriminate better between them.

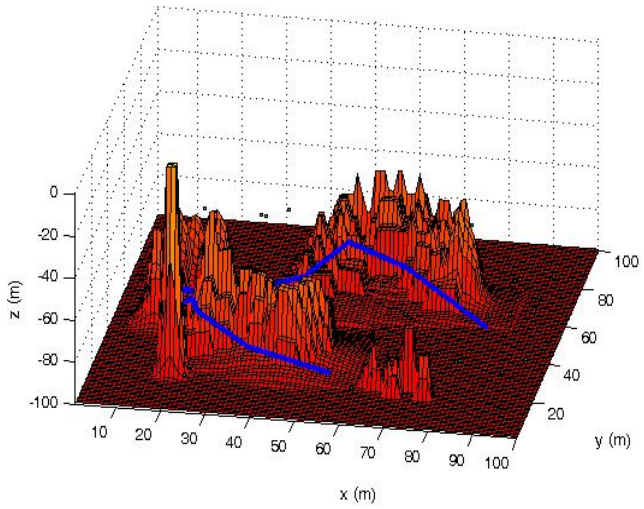
3.2.3 Motion model

For the simulated setup, the motion model is simply the difference between the ground truth positions at time t and at time $t-1$, disturbed with some process noise. The particle state is thus updated considering that value, plus a different noise for each particle, in order to explore more effectively the solution space. For the real setup, the motion estimation is given by the sensors. The Ictineu vehicle is equipped with a SonTek Argonaut DVL unit which provides bottom tracking and water velocity measurements at a frequency of 1.5 Hz. Additionally, an MTi sensor, a low cost motion reference unit (MRU), provides attitude data at a 0.1 Hz rate. These values are integrated in an Extended Kalman Filter (EKF). A 6 DOF constant velocity kinematic model is used to predict the state of the vehicle. Since AUVs are commonly operated describing rectilinear transects at constant speed during survey missions, such model, although simple, represents a realistic way to describe the motion.

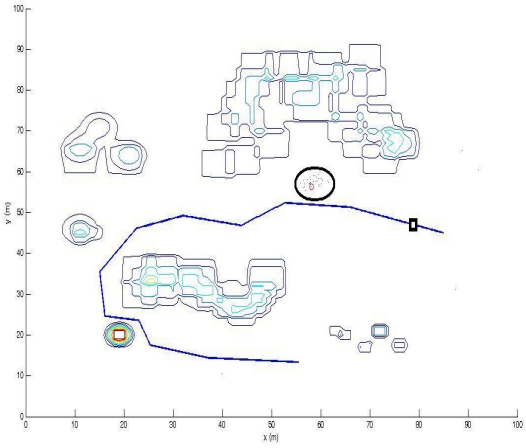
3.2.4 Results

Results on simulated data The first step in the validation of the proposed system is by simulation. Our system can model a vehicle with six degrees of freedom (DOF). In this particular setup pitch and roll of the vehicle are neglected. Additionally, at this point, the sensor's orientation in relation to the vehicle is fixed. A simulated gyroscope is used to have a noisy estimation of the orientation of the vehicle's heading (yaw). A simulated depth sensor provides a noisy estimation of the vehicle's depth. Finally, a simulated sonar is modelled to acquire range profiles, with a field of view of 30m. It is assumed that an *a priori* map of the vehicle's surroundings is known. No assumptions are made on the initial position of the vehicle within the map. The particle state is represented by six variables, three for orientation and three for position of the vehicle, plus an additional variable representing the weight of the particle.

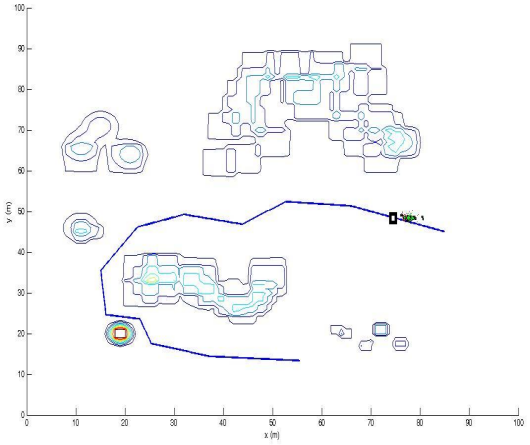
A synthetic environment was created to validate the approach. Different types of



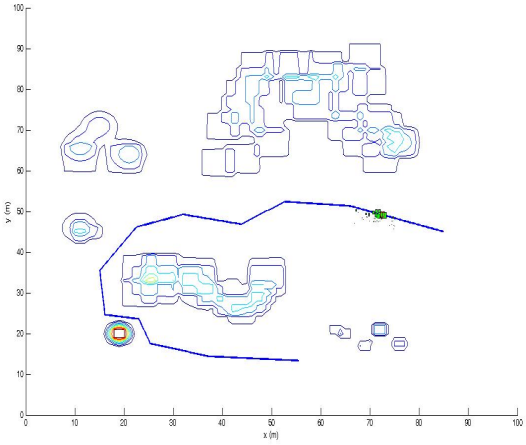
(a) 3D representation of the environment and of the trajectory.



(b) Localisation algorithm starts. The robot is at the small square on the right, whilst the particle converged inside the area delimited by the ellipse.

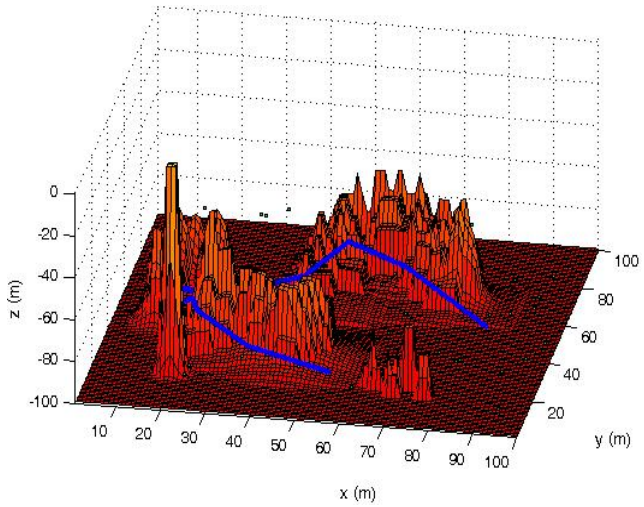


(c) The filter is able to recover from a wrong convergence. All particles are now close to the robot location.

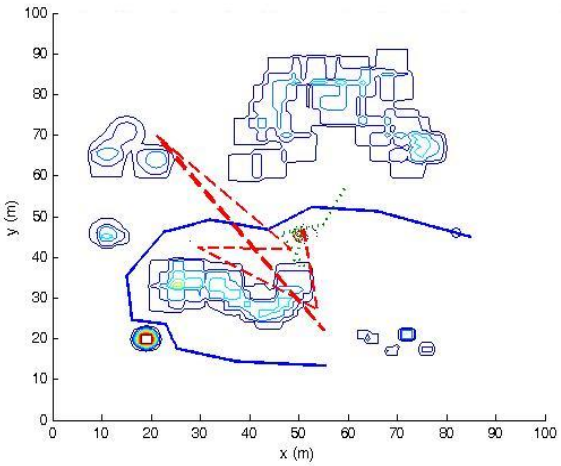


(d) Once converged, position tracking performs well, and the particles are always close to the robot location.

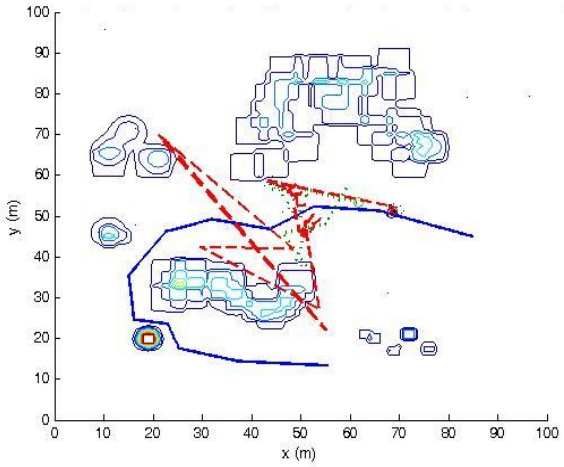
Figure 3.3: Three consecutive states of a mission (2D projection of a 3D simulation). This test shows the ability to recover after a wrong state estimation. The real trajectory is a solid blue line (black in a grey scale image), where the rectangle on top of the line represents the actual position of the AUV at that time; (a) 3D environment and 3D trajectory in blue; (b) Wrong particle convergence: 90% of the particles are in the circle, quite far from the real AUV position; (c) recovering from the wrong convergence: the particles are now close to the real position (with increased likelihood); (d) the actual AUV state has been correctly estimated.



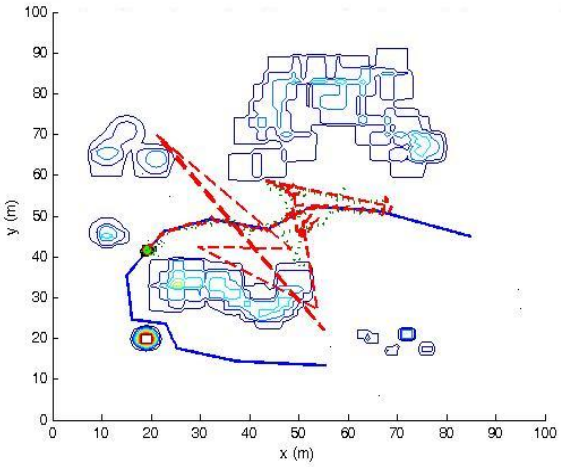
(a) 3D representation of the environment and of the trajectory.



(b) Localisation algorithm starts. The robot is at the small circle on the right, whilst the particle converged towards the centre of the map. The dash-trajectory represents the location of the best particle (i.e. the one with highest weight), while the dot trajectory represents the average on all particle locations.



(c) The filter is able to recover from a wrong convergence. All particles are now close to the robot location.



(d) Once converged, position tracking performs well, and the particles are always close to the robot location.

Figure 3.4: Three consecutive states of a mission (2D projection of a 3D simulation). This test shows the ability to recover after a wrong state estimation. The real trajectory is a solid blue line (black in a grey scale image), where the rectangle on top of the line represents the actual position of the AUV at that time; (a) 3D environment and 3D trajectory in blue; (b) Wrong particle convergence in the centre of the image; (c) recovering from the wrong convergence: the best particle expected position is very near the real position, while the mean expected position is still far, at about the center of the figure; (d) the actual AUV state has been correctly estimated.

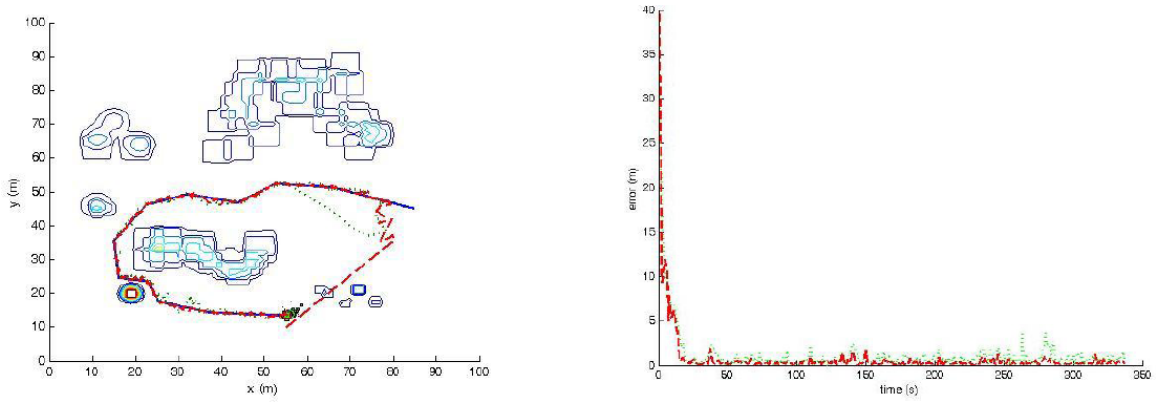


Figure 3.5: Particle Filters for localisation: (left) 2D projection of map, real and estimated trajectories; (right) error in the localisation, both for the trajectory given by the best particle (red, dark gray) and for the mean of the particles (yellow, light gray). The trajectory given by the best particle always performs better, after convergence.

scenarios were considered in order to analyse the algorithm performances. The system has been proven valid for both structured and unstructured scenarios. In more than 96% of cases, the algorithm converges to the real trajectory before the end of the experiment. Figures 3.3 and Figure 3.4 show the ability of the algorithm to recover from a wrong convergence. Two methods have been explored in order to infer the trajectory, given the particles' state. The first method considers the mean of the Monte Carlo approximation (i.e. weighted sum of the particles) and the second one considers the best particle as an estimate of the state of the AUV (a crude estimate of the mode of the distribution). In the simulated setup, the best particle trajectory gives always better results, minimising both the time needed for convergence and the overall error, as shown in Figure 3.5. This setup is the closest to a real scenario with an underwater vehicle: compass and depth/pressure sensors are able to provide a good estimate of those parameters. So, the initialisation of the particles are on two full dimensions (x and y), and on two with a small variance, to account for measurement errors (z and yaw). Increasing the state space to explore with particles is certainly possible, but in that case more particles would be needed. The amount of particles depends on the size of the dimension to explore. For similar dimension, the number of particles is polinomial with respect to the number of dimensions.

Comparison with standard Particle Filter A set of tests varying the number of particles were carried out to define quantitative data and to compare the results with a standard particle filter technique. Three cases were analysed, using the same simulated scenario described in the previous section, with 40 particles, 60 particles and 100 particles. For each case the algorithm runs 200 times and the number of convergence before the end of the trajectory was recorded. Table 3.1 shows the results. Even with a very low number of particles, the algorithm successfully converges to the true robot location before the end of the run in more than 96% of the times.

The same test has been run with a standard particle filter, and results are shown in Table 3.2. Very few runs successfully converge into the true robot location. This is because of the low number of particles which do not represent completely the state space. With a standard resampling step, the state space cannot be further explored and therefore in case no particle is initially close to the initial location, it would be impossible to recover from a wrong convergence.

In order to arrive to similar results than those presented with the proposed algorithm, the number of particles need to be multiplied by a factor of ten, as shown in Table 3.3. Convergence now is much more frequent, because the number of particles covers the full state space much better.

Additionally, a standard algorithm is not able to recover from a wrong convergence, or to solve the kidnapped robot problem, unless there is a cognitive layer that reinitialises the localisation filter.

Results on real data The localisation system was tested on the same dataset used by [94] to perform underwater SLAM. The data was gathered during an extensive survey of a abandoned marina in the Costa Brava (Spain). The *Ictineu* AUV gathered a data set along a $600m$ trajectory which included a small loop around the principal water tank and a $200m$ straight path through an outgoing canal. The data set included measurements from the Imaging sonar (a Tritech Miniking), DVL and MRU sensors. For validation purposes, the vehicle was operated close to the surface attached to a DGPS equipped buoy used for registering the real trajectory (ground truth). Figure 3.6 and Figure 3.7 show the results of the localisation algorithm, in

particles	convergences	percentage
40	193	96.5%
60	194	97%
100	198	99%

Table 3.1: Results on convergence over 200 runs for each configuration. Even with a very low number of particles, the algorithm successfully converges to the true robot location before the end of the run.

particles	convergences	percentage
40	6	3%
60	9	4.5%
100	22	11%

Table 3.2: Results on convergence over 200 runs for each configuration, with a standard particle filter. Convergence is very rare due to the low number of particles, which is not able to cover for the state space.

particles	convergences	percentage
400	162	81%
600	188	94%
1000	200	100%

Table 3.3: Results on convergence over 200 runs for each configuration, with a standard particle filter. Convergence now is much more frequent, because the high number of particles is able to well cover the state space.

two different settings. In Figure 3.6, 100 particles are used and they are spread over an area of $1,848$ square meters. In Figure 3.7, the initial area where the particles are spread increases to $10,368$ square meters and the number of particles is consequently increased to 600.

As it can be seen, the dead reckoning trajectory obtained by merging DVL and MRU data suffers from an appreciable drift (even causing it to go outside the canal). On the other hand, during all the mission, the trajectory computed by the localisation system is very close to the trajectory given by the DGPS. Figure 3.8 shows a zoom in a specific area of the marina (part of the big trapezoid). The particles have different shape according to their weight. It is clear that the error given by dead reckoning trajectory is continuously increasing, while the performance of the trajectory given by the localisation are very good during all the mission.

Comparison with Set Membership approach An approach based on Set Membership was presented in section 2.6. As the algorithm was tested on the same dataset, particle filter results were just shown in the last paragraph, here is a critical analysis of the two approaches, based on the performances. Both techniques are proven reliable and valid for AUV localisation in a man-made underwater environment. It is important to stress that both techniques can work only when the sonar sensor measures are distinctive enough, like marinas and underwater structures. Both do not work well in open sea, if the environment is featureless. Although radically different in the mathematical background, both techniques have common points. Both methods do not make any assumption on the initial position. They are therefore able to perform global localisation and not just position tracking. Both algorithms are very robust

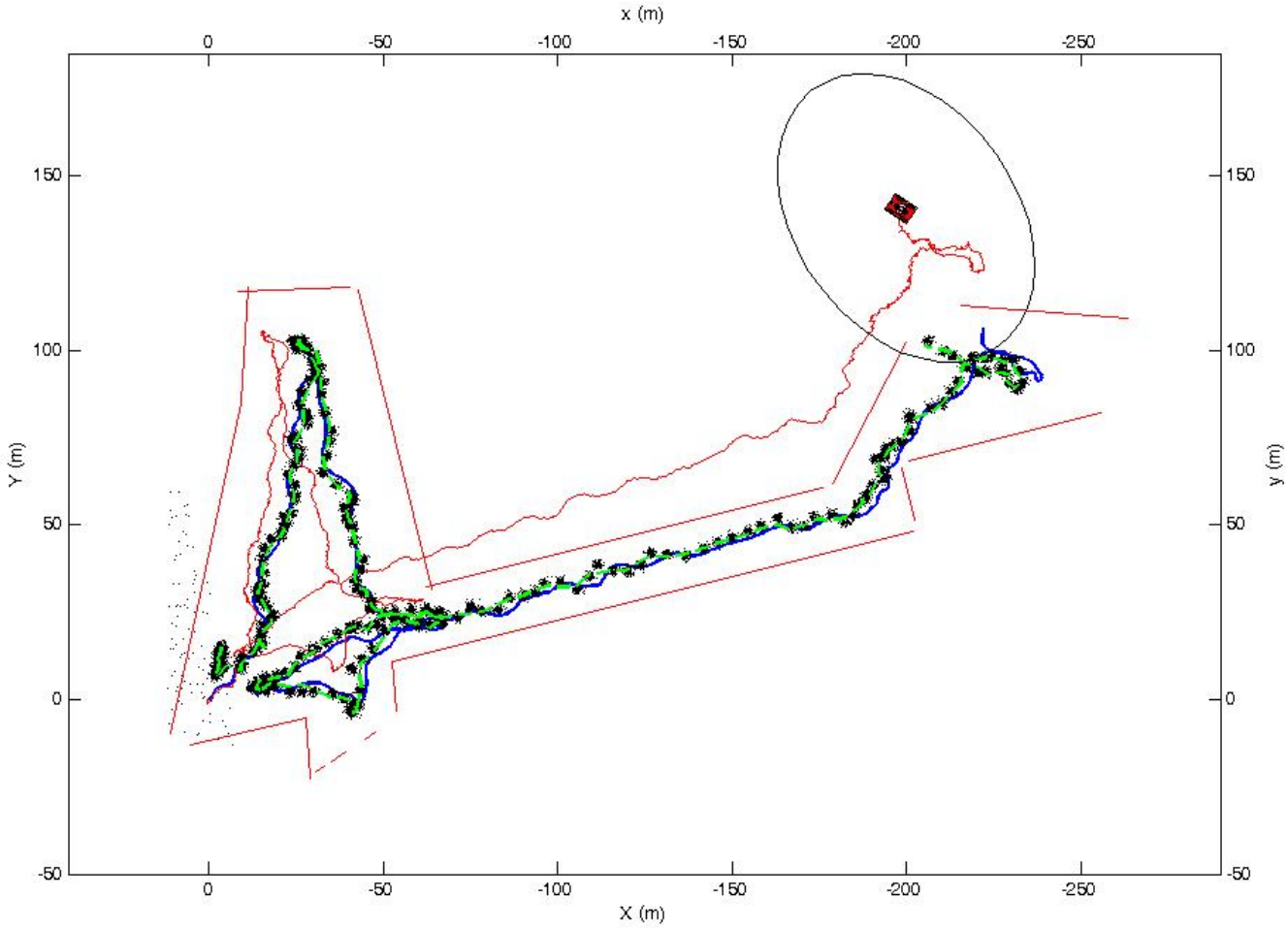


Figure 3.6: A 2D plot of the environment, with the particles, plotted for all the timestamps, the DGPS trajectory (blue), the dead reckoning trajectory (red), the uncertainty ellipse from the dead reckoning, and the trajectory inferred by the particles (green). 100 particles are spread over an area of 1,848 square meters.

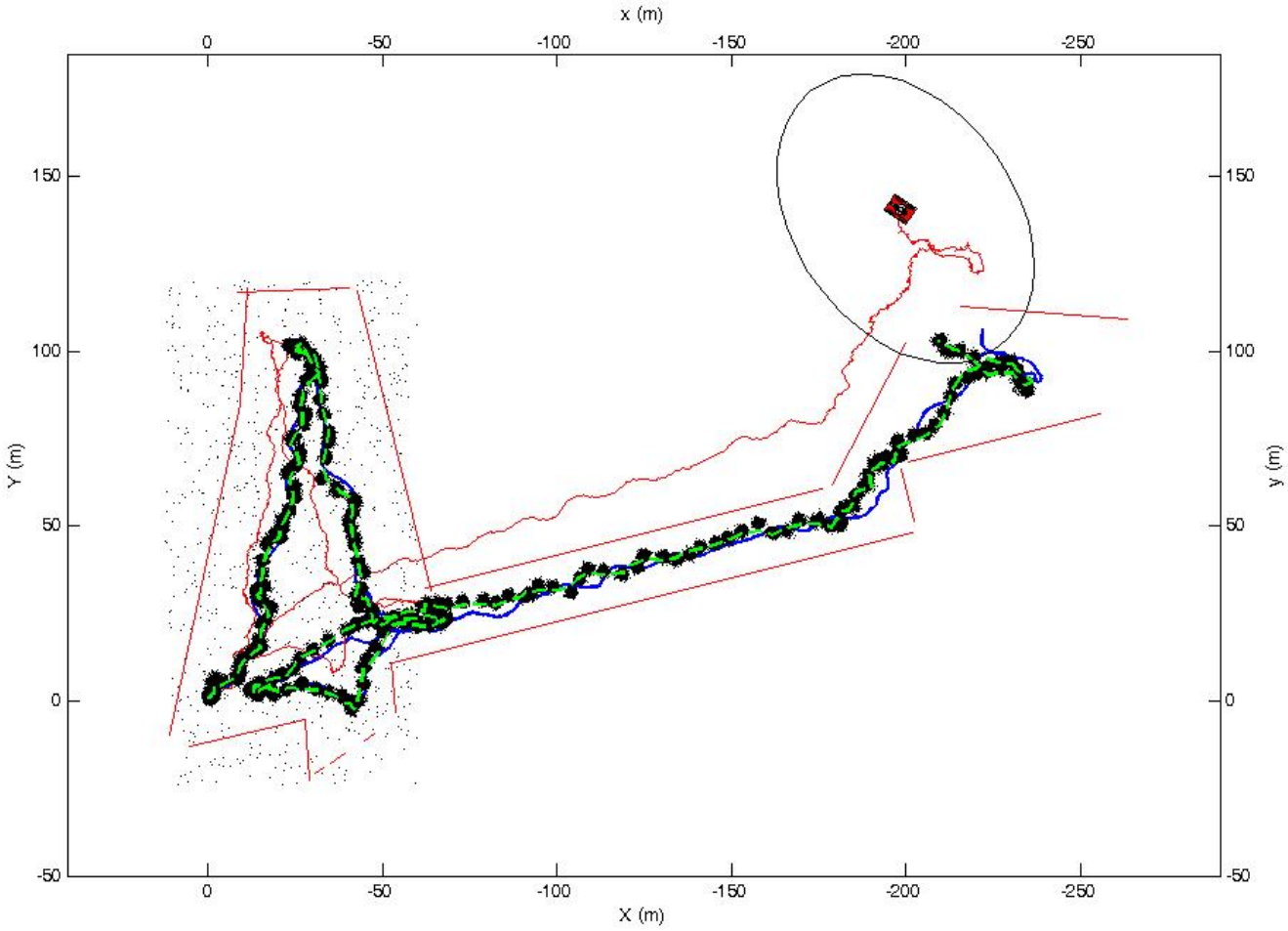


Figure 3.7: A 2D plot of the environment, with the particles, plotted for all the timestamps, the DGPS trajectory (blue), the dead reckoning trajectory (red), the uncertainty ellipse from the dead reckoning, and the trajectory inferred by the particles (green). 600 particles are spread over an area of 10,368 square meters.

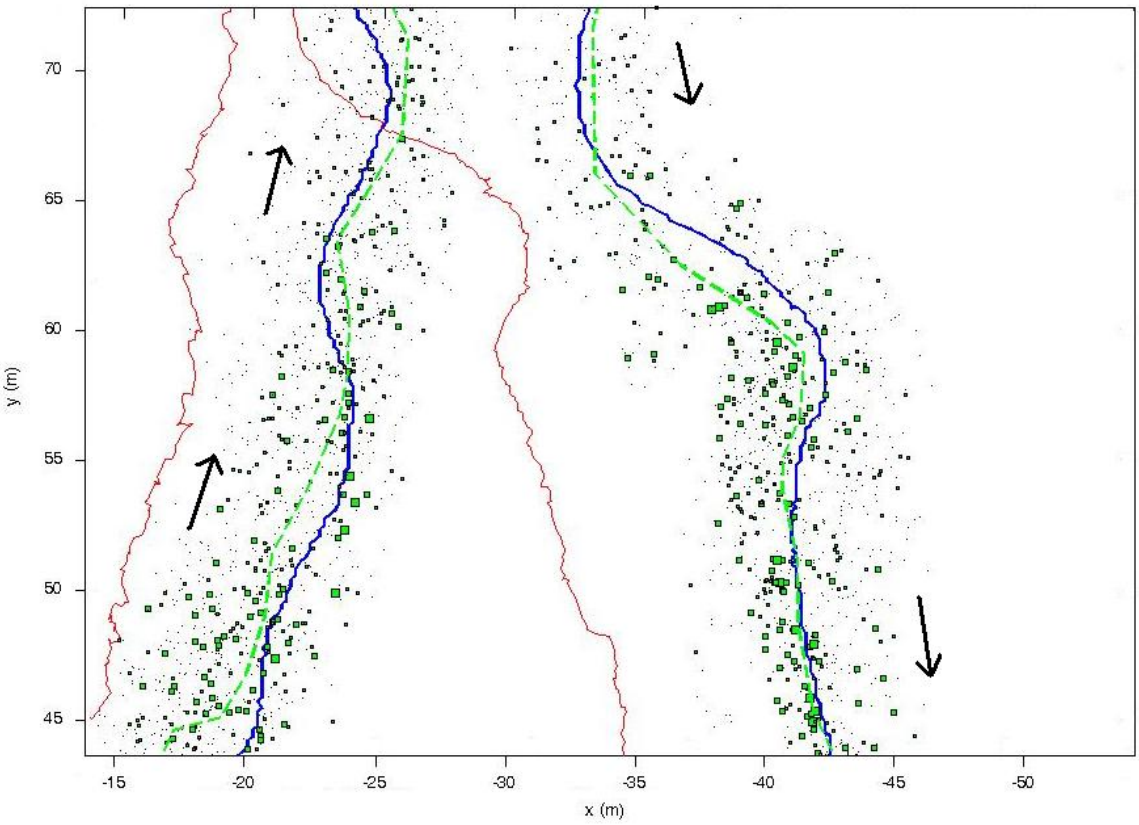


Figure 3.8: A zoom of the area to show how close the inferred trajectory (green - light grey) is to the real trajectory (blue - black) in comparison with the dead reckoning (red - dark grey).

against outliers in the sonar measures. Having an estimation of the AUV motion is helpful in both methods, but it is not necessary. However, for environments which are not very distinctive, it is almost an essential information. In the field trial discussed, it is the case of the corridor. In that area the sensor measure would always return similar values, and an estimation of the motion is therefore necessary. Both methods can be easily parallelised. The computation of the simulated sensor measure from each particle represents the most expensive part for the Particle Filter algorithm. However, each particle is independent and its associated sensor measure can be computed in parallel, or precomputed and stored in a hash-table. The simulated sensor measures could also be precomputed for all possible position, and the information retrieved with an hash table, thus speeding up the process. Again, the weight calculation for each particle can be computed in parallel. As for the set membership approach, the search space which is initially represented by a box is usually divided into smaller boxes which can be processed separately. The process of characterisation of each box i.e. the process of associating the number of consistent measurements with a position box is in fact independent for every box. Both algorithms are able to recover from wrong convergence and inconsistent situations. The particle filter approach is able to recover *dynamically*, i.e. without changing state of the algorithm. Through the *resampling* step, a portion of random particles is generated, thus allowing a wide exploration of the environment and the recover from a wrong convergence. The Set Membership approach can detect inconsistent situations and perform again global localisation, called from the author *static* recovery. The algorithm state changes to global localisation and, when a convergence is reached, changes back to position tracking.

Particle Filter Techniques can handle non Gaussian and non linear processes. Set Membership Techniques can also handle non linear processes. Set membership techniques do not directly involve probabilistic distributions. Those methods require assumptions on the membership of the variables of the problem. If the assumptions are correct so is the solution. Knowing the probabilistic distribution of a variable can be useful to make assumptions on its membership. As an example, consider a variable x following a Gaussian distribution $N(m, \sigma)$ and x_0 a measurement of this variable. It is possible to assume that the real value of x variable is included in the 99% likelihood interval $[x_0 - 3\sigma, x_0 + 3\sigma]$. Note that in the 1% case where that assumption is not true the measurement x_0 can be considered as an outlier and is taken into account by the robust algorithm, as outlined in [100].

A clear disadvantage for both methods is represented by the computational requirements. However they are both feasible for real-time execution and they have been used integrated in the AUV architecture. The field trials in the Marina took about 1 hour to be performed. The Particle Filter algorithm has been tested postpro-

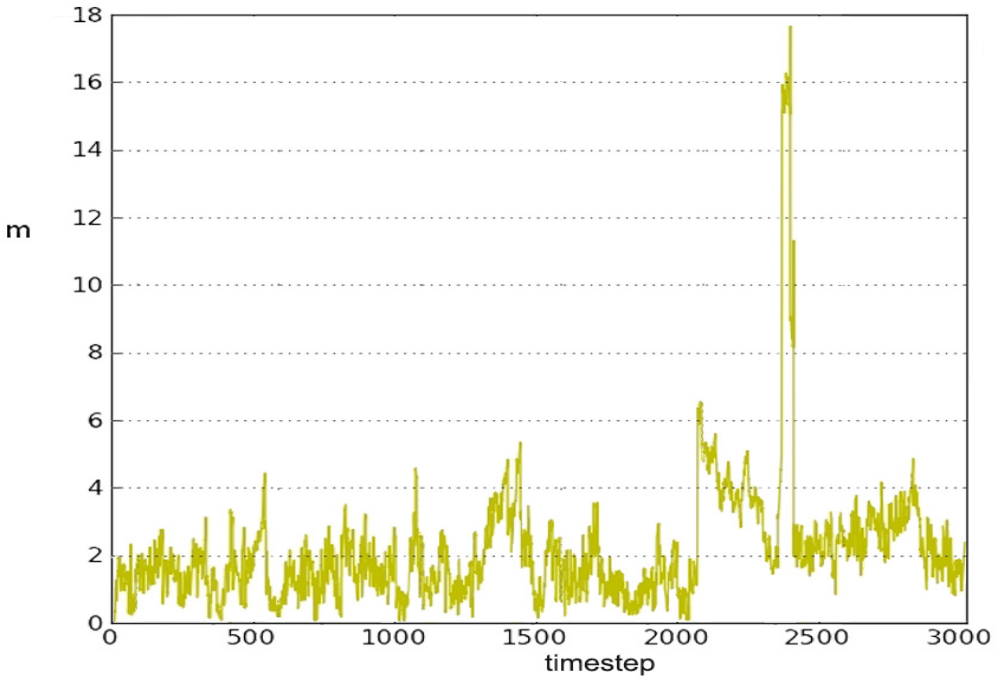


Figure 3.9: Error plots for the Set Membership Approach. The high peak is determined by the vehicle being in the central corridor. That situation cannot be handled in a robust way by this technique.

cessing the data, with 31 minutes needed on Matlab, on a Core2 2.2 GHz. The Set Membership approach was implemented using C/C++ and needed 1hour 10minutes to execute on one core of a Centrino duo T2500 at 2GHz.

Figure 3.9 and Figure 3.10 present the error plots of the two algorithms. The green lines is the error of the Set Membership approach, whilst the black line is the error of the Particle Filter approach. It is possible to see that the error is less than $2m$ in 80% of the cases, which is good compared to the length of the trajectory ($> 600m$). In the case of Particle Filters, it is normal that the initial error is slightly bigger, because global localisation has been performed initially. It is also to be noted that despite some noisy measures increased the error in some cases, the algorithm shows the possibility to gradually correct itself and reduce the error, recursively estimating the vehicle state. Figure 3.11 shows the two error plots in the same figure. Analysing that together with the vehicle's trajectory, it is possible to understand better the sources of error. In both cases, spikes of error appear when the data from the sensor are not good enough, for example when the vehicle is in the central area, with few features captured by the sonar, or are not distinctive enough, for example when the vehicle is in the corridor.

Considering that the main disadvantage of the Set Membership approach is that

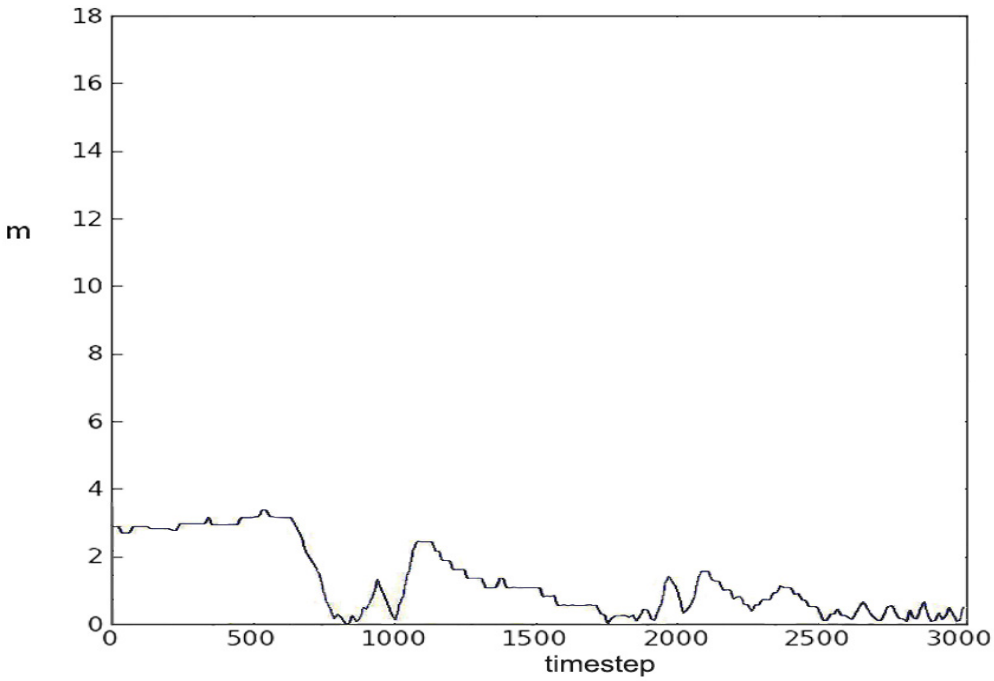


Figure 3.10: Error plots for the Particle Filter Approach.

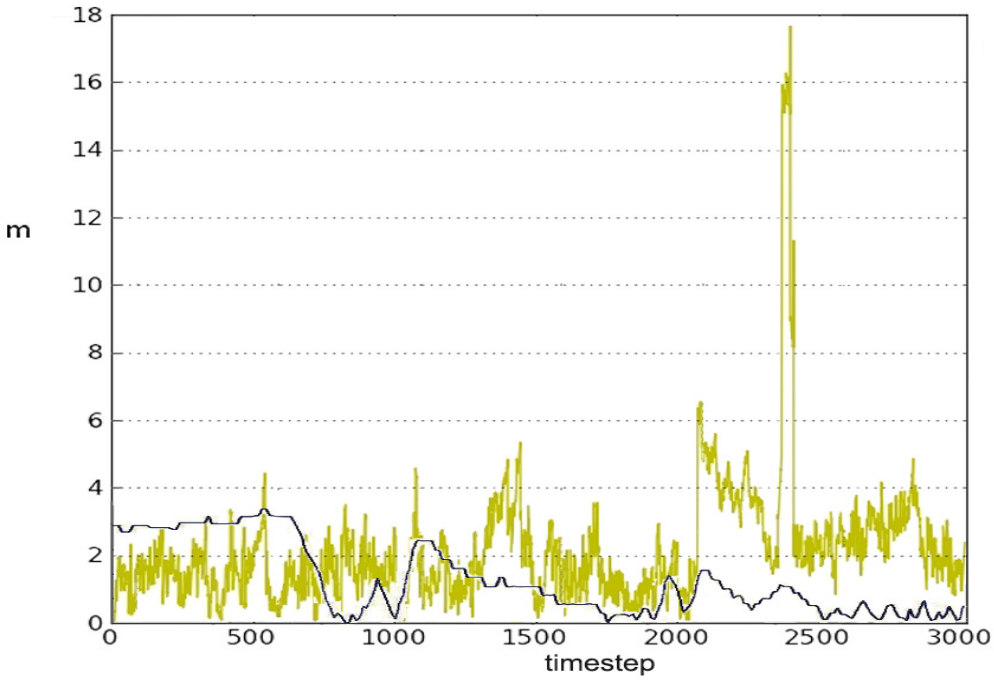


Figure 3.11: Comparing error plots among Particle Filter and Set Membership approaches.

the solution is often given in a form of large sets for positions, without knowing where is the biggest probability of occurrence, a possibility to make use of the best features of both approach would be to use a Set Membership approach to find an overall guaranteed position and then to apply a probabilistic approach, like Particle Filters, to define the uncertainty and to calculate the most likely state.

3.3 Partially known maps

3.3.1 Particle Filters with partially known map

Most of the available techniques consider the problem of localisation in a map previously known by the robot. Neither corruption nor incompleteness of the map is usually considered, unless the research is about the full SLAM problem. However the assumption to have a perfect map of the environment is often unrealistic. It is more common to have an estimation of the map or to have an incomplete map, with most features known by the robot, but with some differences between the sensed map and the previous knowledge. For a mobile robot moving in an indoor environment, the previously known map could be, for example, the walls of the building. It is quite unlikely that the robot can be aware of the position over time of a chair in a room, unless its map is continuously updated. In the underwater domain the same situation appears. Water currents can change the natural profile over the years, animals and other floating objects can interfere with the sensors, the position of subsea man-made installations can change with time from the deployment point and maps could be not fully updated. Moreover, in tank tasks, the tank walls can be previously known, but it is often impossible to have a detailed map of other objects/vehicles/people in the tank itself. In the scenario of the SAUC-E competition 7.1.1, for example, the dimensions of the tank are known, while position and orientation of different objects in the tank are unknown. In all these situations, a full-SLAM approach helps, but it is more computationally expensive than a localisation one. The goal of this section is to explore the algorithm presented in the section 3.2 in various condition when the map is not completely known.

Experimental Results

For these tests, the Cartesian robot in the OSL tank was used, mounting the Tritech SeaKing on it. Three different fields of view were tested, in order to study the error in function of the portion of the environment which can be observed. The operating environment was a rectangular tank, 4 metres long, 3 metres width and 2 metres deep. The sensor data were scaled by a factor of 25, in order to simulate

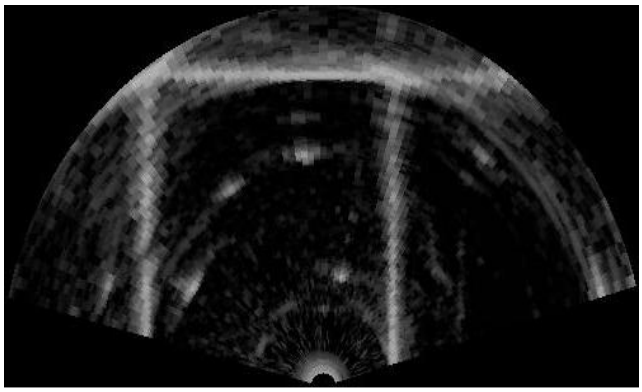
a bigger environment. Four irregular mid-water objects were added in the pool, but the robot was only aware that it was in the rectangular pool. In this way the problem has moved from localisation in a completely known map to localisation in a partially known map. In Figure 3.12 there is an example of the sonar image, of the processed image with range values for each angle and the simulated image from the same position.

As shown in the figure, the simulated and the real picture are different. That is because the vehicle is not aware of the mid-water objects. The results presented in this section show the robustness of the proposed algorithm and its capability to be used in real missions, being able to localise and navigate around underwater structures.

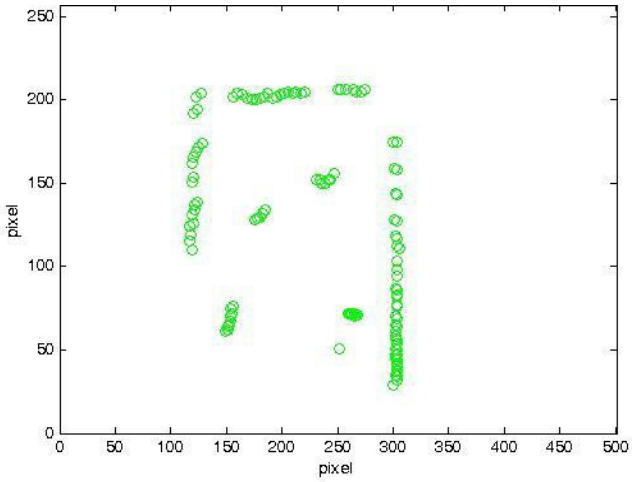
Figure 3.13 shows the results of the algorithm. The real trajectory is plotted in blue (black). The expected trajectories, given by particle analysis are plotted in green (light grey) and in red (dark grey). The green dash trajectory is given by the mean of the particles and the red dot trajectory is given by the best particle. As the figure shows, at the beginning, the inferred trajectories are not close to the real trajectory, because no assumptions are made on the initial position of the vehicle within the map. After a short time, the particles converge near the real position and they do not lose it. The figure also shows the particle distribution at the last step, plotted according to their weight. Particles near the real trajectory are bigger than particles with a low weight.

Figure 3.14 shows the error between the real trajectory and the trajectories inferred by the particles. The green (light grey) dash error line is referred to the trajectory given by the mean of the particles, whilst the red (dark grey) dot error line is referred to the trajectory given by the best particle. The considerations about errors detailed for the synthetic environment are not valid any more. It is still true that the best particle trajectory converges faster than the mean particle trajectory. However there is not much difference between the two errors. While in simulation it was clear that the best particle trajectory was to be preferred, using the real data, over more than one thousand tests with different configurations, they were substantially equivalent. A possible explanation lies on the fact that the real tests are performed in a partially known environment and not in a fully known one. To analyse the performance of the system, the number of particles used was changed among 20, 40 and 80 particles. The field of view of the sonar was also changed among 50, 100, 150 degrees. For each configuration 100 different tests were run. Figure 3.15 shows the average error over the total of 900 runs of the algorithm, 100 for each of the nine different configurations.

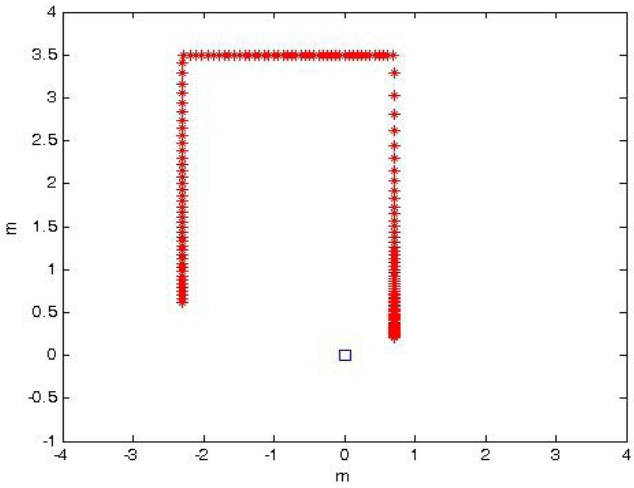
The error varies from 8 to 60 cm, depending on the number of particles and on the field of view of the sonar. It is quite important to notice that, after the first iterations, the error is given only by the x component of the particles, as in 100of



(a) Sonar image, from the Tritech SeaKing, mounted on the Cartesian robot.



(b) Segmented sonar image, extracting a distance value for each beam.



(c) Simulation of the sonar data from the same location. The vehicle is not aware about the four buoys.

Figure 3.12: Differences between the perceived reality and the vehicle's knowledge. (a) Real sonar image; (b) segmented sonar image, extracting distance value for each beam; (c) simulated sonar image from the same location. In addition to the noise, the main difference is in the four mid-water objects, of which the vehicle is not aware of.

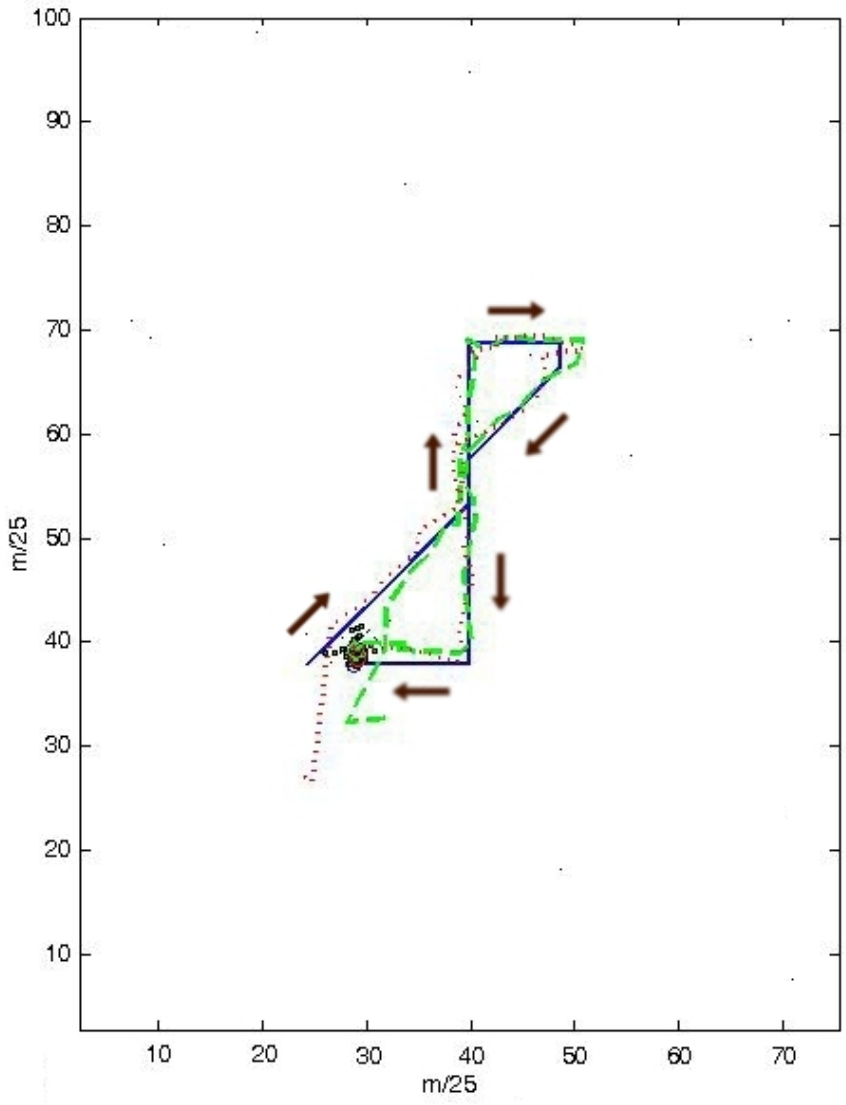


Figure 3.13: Real results with the Cartesian Robot: a 2D plot of the environment, with real trajectory (blue) and expected trajectories, given by particle analysis. The green (light gray) dot trajectory is given by the mean of the particles, while the red (dark gray) dash one is given by the best particle. The real trajectory starts at the beginning of the blue (black) line, on the bottom left of the figure. The particles in their last configuration are also shown, at the end of the trajectories, on the bottom center of the figure.

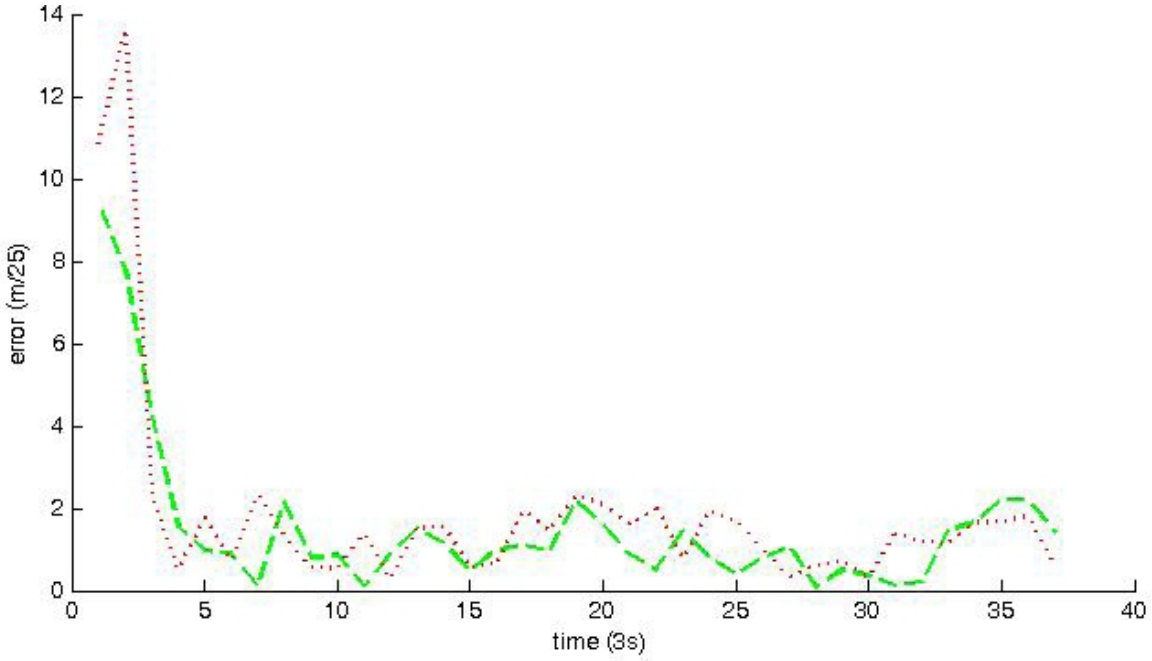
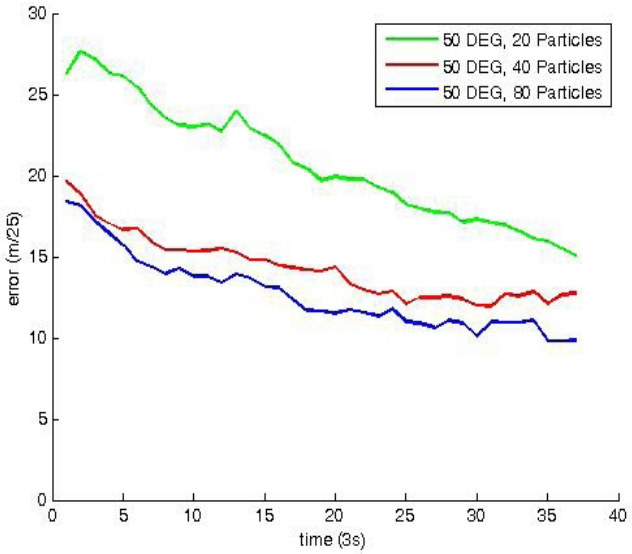


Figure 3.14: Real results with the Cartesian robot: error between real trajectory and expected trajectories, inferred by the particles. The red (dark gray) dash error line is given by the best particle trajectory, while the green (light gray) dot error line is given by the mean trajectory.

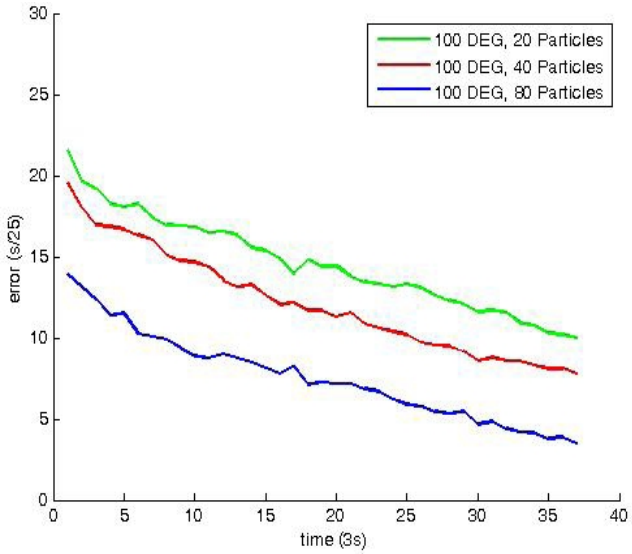
tests the particles converged to the real y value. With limited field of view and with the high noise, this fact is understandable. Looking at the error plots, it is clear that the precision of the final solution is directly linked to the number of particles and to the field of view. Within certain limits, they can compensate each other. However solutions with very small field of view are in general very imprecise, as well as solutions using too few particles.

3.3.2 EKF Localisation for a partially known map

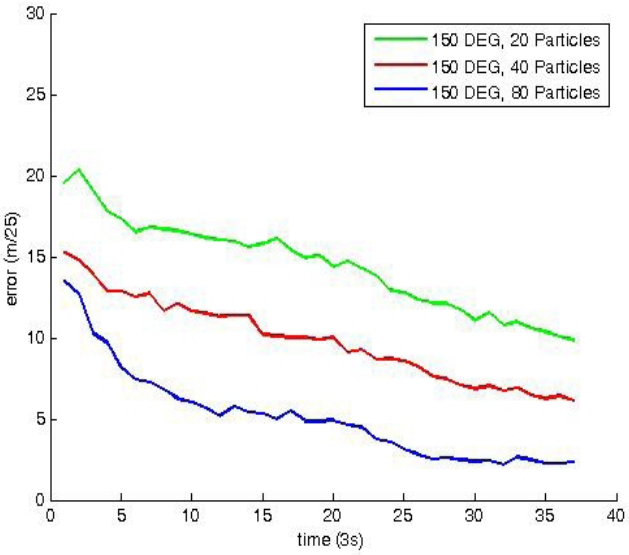
This method uses the EKF localisation approach for a map defined by known and unknown natural landmarks. The state, motion, prediction and update models remain the same as the EKF localisation for a map of known landmarks, presented in section 2.4. However, a different Feature Matching algorithm is proposed. Observations are made using imaging sonar that scans the horizontal plane around the vehicle. The observation consists of a relative distance and orientation from the vehicle to the feature. Point features are extracted from the scans and are matched against existing features in the map. This feature matching is different for a map comprising known and unknown landmarks in the map. When a new range and bearing observation is received from the feature extraction process, the estimated position of the known feature from the predicted position of vehicle is computed. This position is then



(a)



(b)



(c)

Figure 3.15: Real results with the Cartesian Robot: average error over 100 tests for each configuration. Each figure represents a different field of view ((a) 50 deg, (b) 100 deg, (c) 150 deg) and for each figure 3 different number of particles are represented (20, 40, 80 particles)

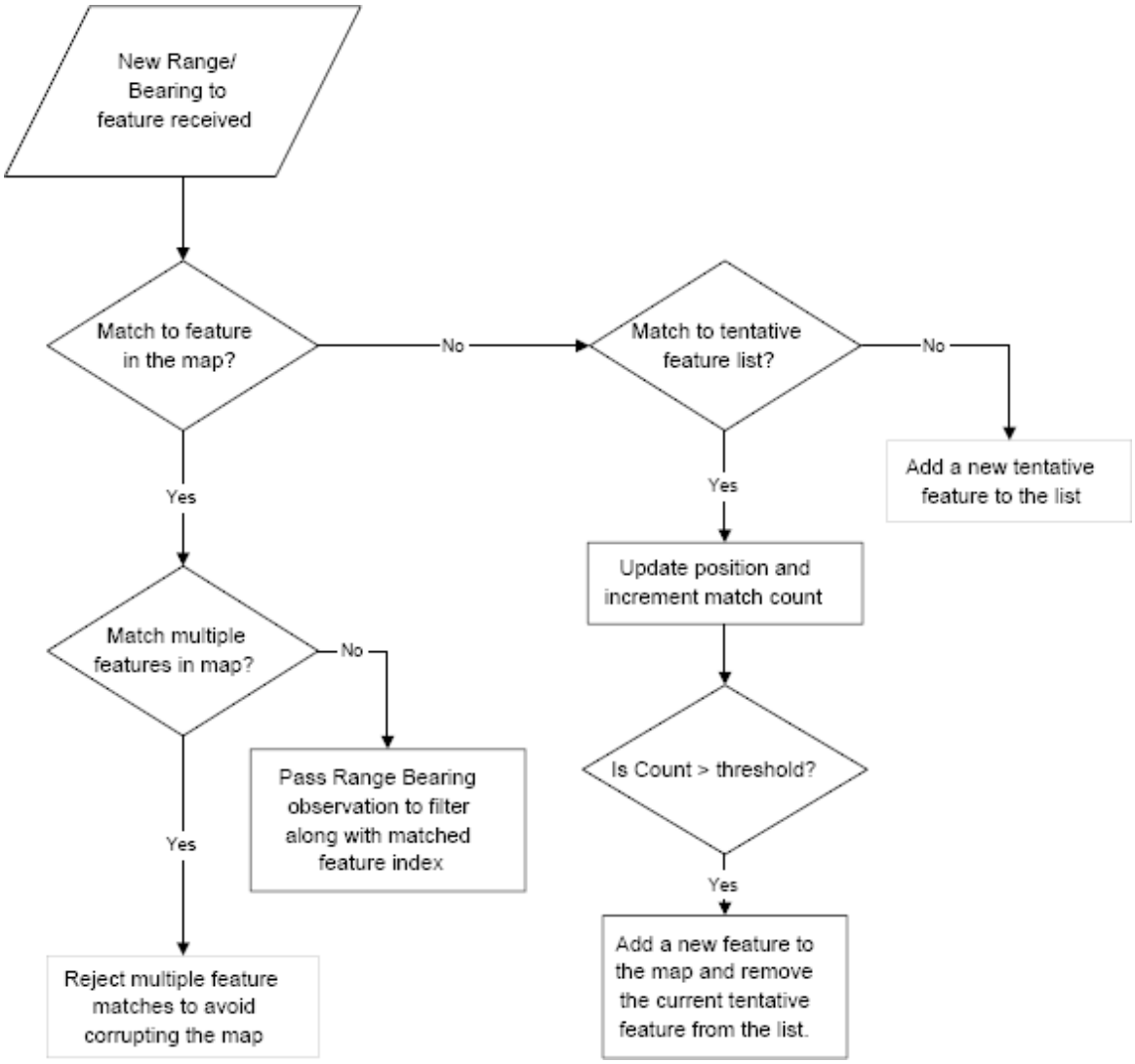


Figure 3.16: The Feature Matching Algorithm

compared with the estimated positions of the features in the map using Mahalanobis distance. If the observation can be associated to a single feature, the EKF is used to generate a new state estimate. An Observation that can be associated with multiple targets is rejected since false observations can significantly harm the integrity of the estimation process. Figure 3.16 shows the matching process. Similarly, if the observation does not match to any targets in the current map, it is compared against a list of tentative targets. Each tentative target maintains a counter indicating the number of associations that have been made with the feature as well as the last observed position of the feature. If a match is made, the counter is incremented and the observed position is updated. When the counter passes a threshold value, the feature is considered to be sufficiently stable and is added to the map. If the potential feature cannot be associated with any of the tentative features, a new tentative feature is added to the list. Tentative features that are not re-observed are removed from the

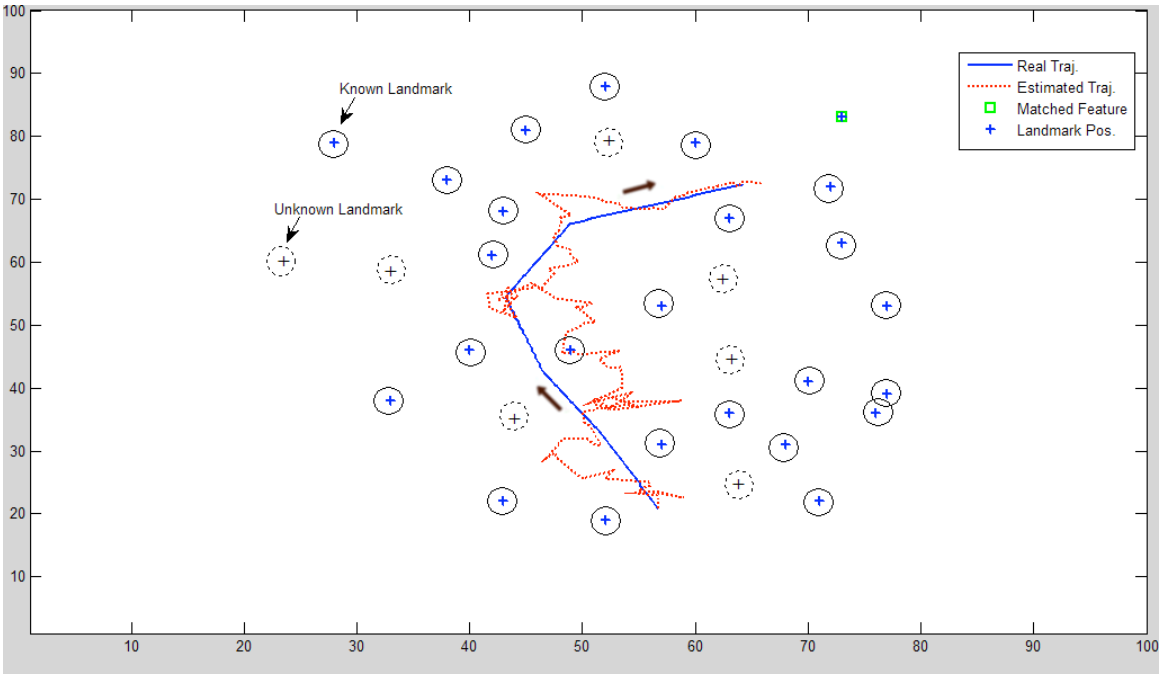


Figure 3.17: EKF for partially known map - Real and Estimated Trajectory

list after a fixed time interval has elapsed.

Numeric results

In this case, the map is represented by a mixture of known and unknown narrow long pillars placed at the bottom of the underwater environment. In a similar approach with EKF for known Landmarks, the estimated trajectory, error and covariance in the estimate of the vehicle state is shown in Figure 3.17 and in Figure 3.18.

The increase in error and covariance in the vehicle state, however, can be corrected using the EKF. In the update model of EKF, the distance from the predicted position to each of the known landmarks is computed and a feature matching is performed to match the observed landmarks to one of the known features in the map. Consequently, the innovation and covariance between this landmark and real measurement to a feature is computed. The distance in the feature matching stage is computed in the Mahalanobis sense. Finally, the Kalman gain is computed from the innovation covariance and uncertainty of the prediction stage (i.e. motion model) and the state of the vehicle is consequently corrected. Figure 3.18 shows that there is a higher error in the update model when no match is found for the observed features. When a match is found the update model of EKF is used and the error and uncertainty decreases. As compared to previous implementation of EKF, in this case there is a higher level of error. The same holds for the uncertainty. This implementation is further geometrically constrained by the fact that the landmarks shall be narrow, so that the contour

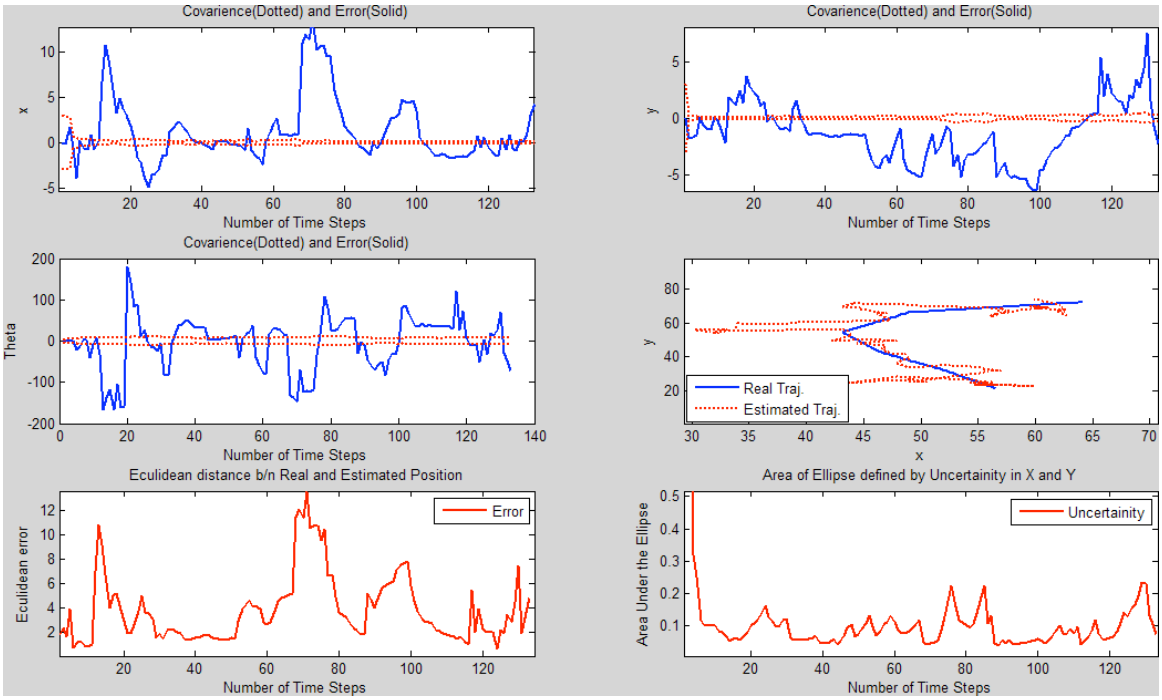


Figure 3.18: Experimental results for EKF localisation on a map of mixture of known and unknown landmarks.

of the landmark would be closer to the centre of the pillar. This will help to reduce the error associated with the feature extraction stage of the measurement model.

3.3.3 Conclusions

In this section the noise and the possibility of an incomplete map has been presented. The particle filter approach was able to cope with those quite well, and results have shown the performances with varying field of view and number of particles. A modification for the data association algorithm is presented for the case of Extended Kalman Filters, with the possibility of modifying the map, adding new information, without solving the full SLAM problem, computationally more expensive.

3.4 Particle Filters merged with EKF

Both Extended Kalman Filters and Particle Filters have strong points and weak points. A combination of the two filters is presented in this section, in order to handle the limitations in using only one of the two. The main strong points of the Particle Filters approach are the possibility to start without an estimation of an initial position and to be able to recover from wrong convergence. The main strong point of Extended Kalman Filters is the computational efficiency, compared with the previous solution. The main weak point of EKF, in addition to needing an estimation of the initial

position, is the impossibility to assure recovery from a wrong convergence. In cases of high-nonlinearities, additionally, the Kalman filter solution is not very reliable and thus a correction through particle filtering can substantially help the state estimation. In order to handle these problems and to maximise the strong points of both methods the following algorithm is proposed:

- localisation starts with an improved Particle Filter approach, as there are no assumptions on the initial position of the vehicle;
- once a convergence is reached, the system switches to a Kalman filter approach, giving as input the estimated state by the Particle Filter;
- the control is given back to the Particle Filter module in these cases: (a) after a fixed period of time, to check if the convergence is right and, if needed, to correct the state estimation - *routine procedure*; (b) the covariance matrix of the EKF grows over a fixed threshold, meaning a high imprecision - *emergency procedure*.

When the Particle Filter module is called because of a routine procedure, the particles are initialised as a normal distribution, centred in the expected position given by EKF module. When it is otherwise called by the emergency procedure, the initial distribution is a mixture of a Gaussian distribution centred, as before, in the expected position given by EKF module and a uniform distribution over the map. In this way, the process evolution is taken in account, but on the same way the uniform distribution helps to explore the whole environment, because the module was called after a mislocalisation process has been detected. This method has been tested successfully and shows efficiency, accuracy and robustness.

Naturally an important role is played by the landmarks. Defining landmarks as long and narrow pillars will reduce the error in the feature extraction stage and improve the localisation results. Similarly the higher the number of the landmarks, the better the localisation results.

3.4.1 Numeric Results

Our system can model a vehicle with six degrees of freedom (DOF), plus a DOF for the mounting of the sensor on the sway axis. In this particular setup we have assumed that pitch and roll of the vehicle are neglected. Additionally, at this point, the sensor orientation in relation to the vehicle is fixed. A simulated gyroscope is used to have a noisy estimation of the orientation of the vehicle (yaw). A simulated depth sensor provides a noisy estimation of the distance between the vehicle and the seabed. Finally, a simulated profiling sonar is modelled to acquire range profiles. It is

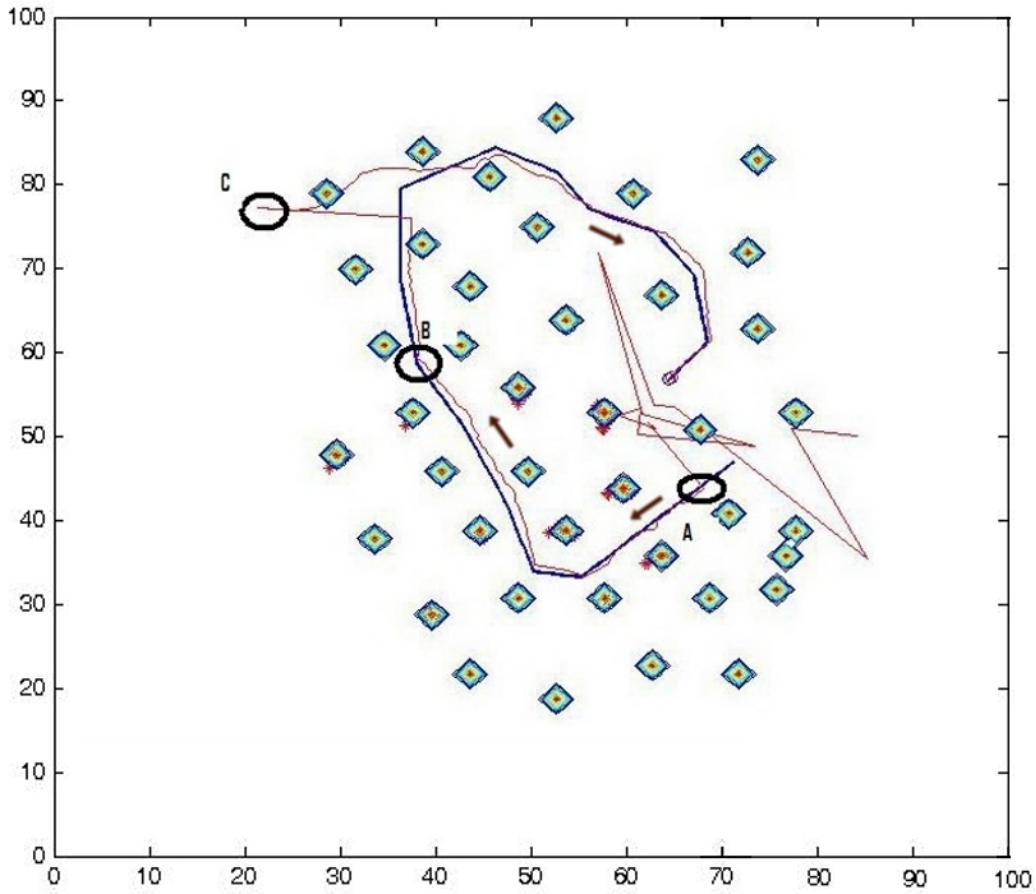


Figure 3.19: A 2D plot of the environment, with real trajectory in blue (black) and expected trajectory in red (gray). The starting point is on the bottom. (A) End of Particle Filter Module, with an estimation of the position given to the EKF Module; (B) routine procedure: the particles correct the EKF estimation; (C) emergency procedure: the Particle Filter Module is called because a significant growth of the uncertainty was detected.

assumed that an a-priori map of the vehicle's surroundings is known. No assumptions are made on the initial position of the vehicle within the map. The particle state is represented by six variables, three for orientation and three for position of the vehicle, plus an additional variable representing the weight of the particle.

In Figure 3.19, the results of the combination of Particle and Kalman Filters are presented. A 2D projection of the 3D environment is shown for more clarity. The real trajectory is plotted in bold blue (black) and starts from the bottom. In Figure 3.20 the error between the expected trajectory and the real one is plotted.

At the beginning, as no assumptions are made on the initial position, the error is quite high and the expected trajectory not very stable. At point A, the particle filter algorithm converges and gives the control to the EKF module. The EKF run until a bit before point B, when it is stopped by the controller (routine procedure). In B the

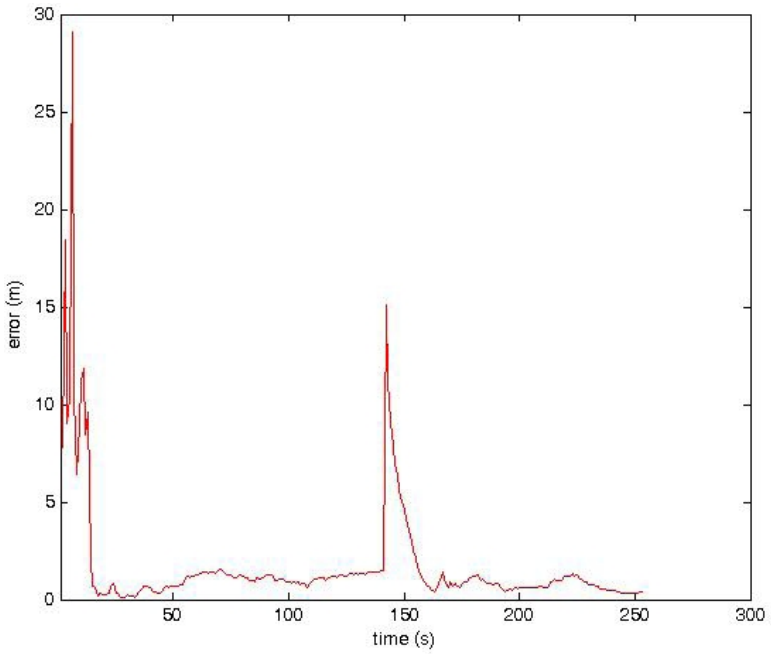


Figure 3.20: Error between real trajectory and expected trajectory. As soon as the particles converge, the error goes close to zero, as well when they are called to correct the estimation.

particle filter module corrects the expected position given by the EKF module and gives back the control. At this point, a mislocalisation is simulated, with the expected position going to point C and with a grow in the covariance matrix. The emergency procedure is thus called. The particle filter module takes the control of the localisation and corrects it in a few steps. The ability to recover from wrong convergence is a key point, very important to increase vehicle autonomy. It has to be noted that the output of the initial run of the Particle Filter module does not necessarily need to be very close to the real position. It is very unlikely, as it has happened only once in 70 tests. However, it is interesting to see that the Kalman Filter Module is able to correct the initial not accurate estimation given by the Particle Filter Module. An example is shown in Figure 3.21.

3.4.2 Experimental Results

For this localisation approach the four mid-water objects in the pool represent the landmarks detected and associated by the EKF module. Figure 3.22 shows the results of the algorithm. The real trajectory is plotted in blue (black). The expected trajectory is plotted in red (dark grey). The expected trajectory is given at first by the Particle Filter module and, soon after the convergence, by the EKF module. After the first steps, in which the Particle Filter module is determining the vehicle position, the trajectory becomes smoother and is close to the real trajectory.

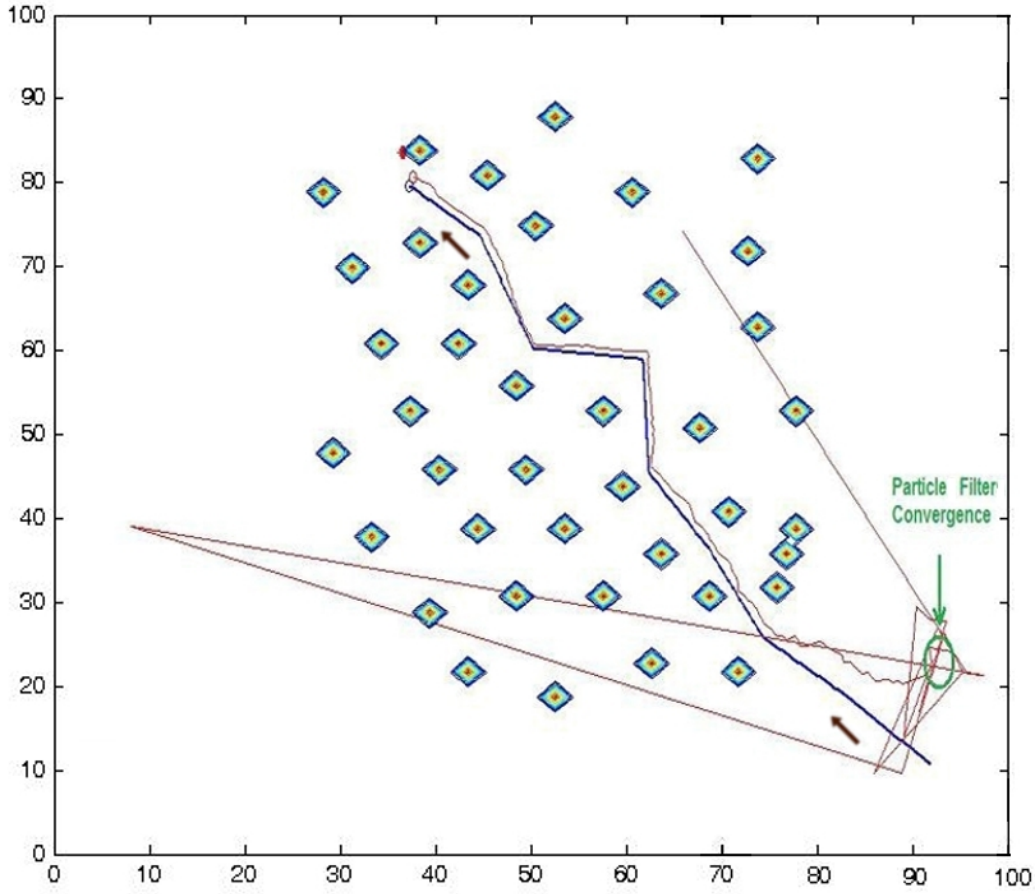


Figure 3.21: A 2D plot of the environment, with real trajectory in blue (black) and expected trajectory in red (gray). The starting point is on the bottom. The initial Particle Filter Module ended with a very noisy estimation of the real vehicle. However, the EKF module corrected it in a few steps.

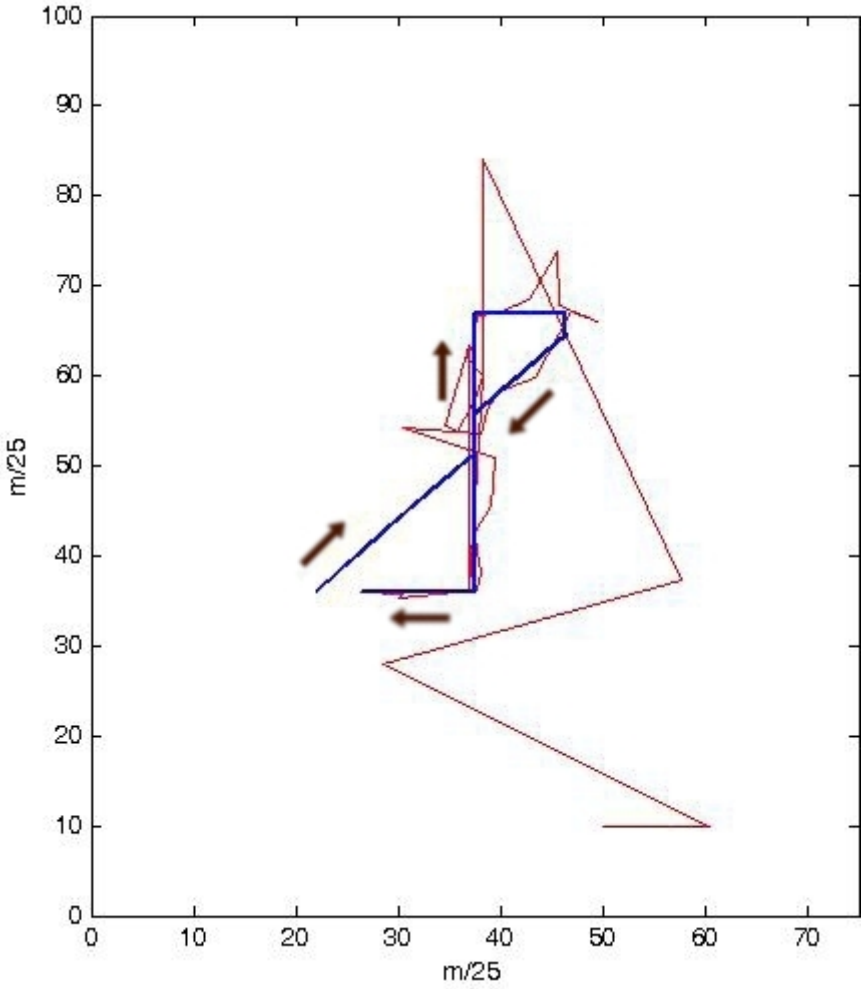


Figure 3.22: Extended Kalman Filter and Particle Filter: tests in the OSL tank. The real trajectory of the vehicle is plotted in bold blue (black) and the expected trajectory is plotted in red (grey).

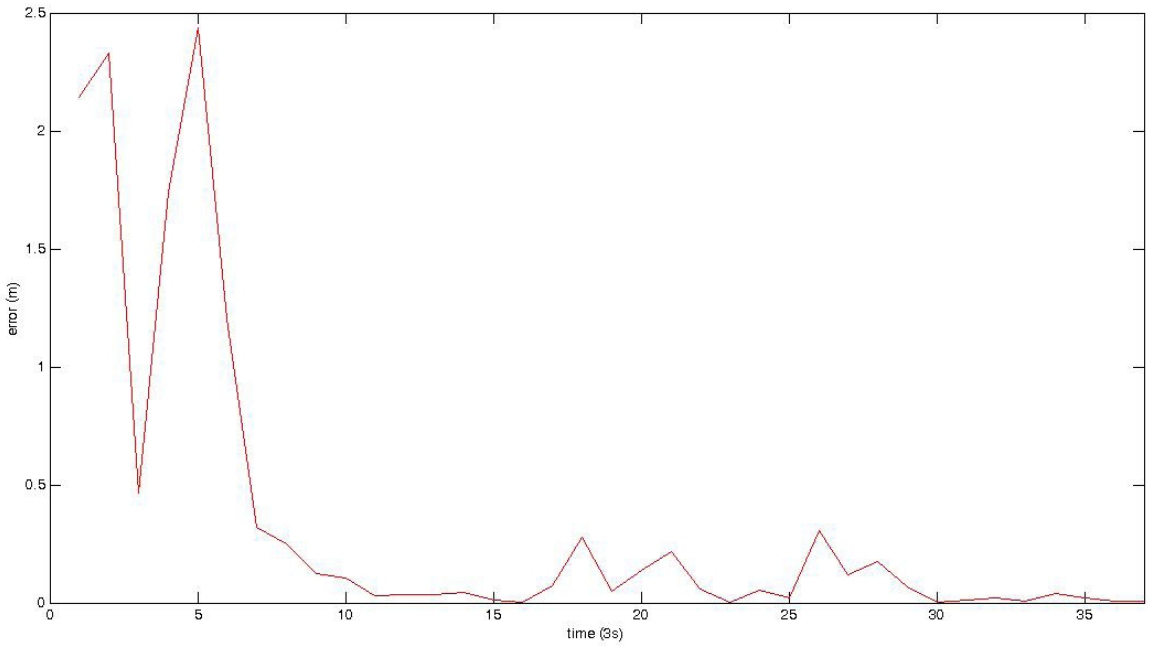


Figure 3.23: Plot of the localisation error.

Figure 3.23 shows the error between the real trajectory and the trajectory inferred by the system. The results are very good, as the error level, after the particle convergence, is always lower than 30 cm. Performing the algorithm over 200 tests, the error was always lower than 40 cm and lower than 15 cm in 89% of the cases. Comparing the results given by the Particle Filter module and those given by the integrated module with the Extended Kalman Filter, it is possible to say that both present very good results. The Particle Filter algorithm is clearly more accurate, but in systems where computational power is a sensitive issue, the integration with the Kalman Filter provides a very good trade-off between accuracy and speed.

3.5 Conclusions

This Chapter has presented a contribution in the field of autonomous robot localisation using passive techniques. After the literature review and the analysis of the state of the art presented in the previous Chapter, the focus went on novel techniques in the field of Particle Filter and Kalman Filter.

An improved particle filter algorithm, adapted for the underwater domain, was therefore presented and compared with state-of-the-art techniques, showing reliability and efficiency.

Then the topic of partially known maps was analysed, with results both using Particle Filter and Extended Kalman Filter.

Whilst particle filters are chosen because able to handle multiple hypotheses, and

are not limited to linear and Gaussian processes, they are certainly more computationally expensive than EKF. Thus an approach to use both algorithms was presented, in order to *intelligently* switch between them according to the current state, goal and circumstances.

Particle filters (with the possible extension with EKF) are considered the main filtering technique explored in this thesis, and will be the base for the *active* localisation system presented in Chapter 6.

None of the presented approaches consider the control of the vehicle in the loop to facilitate the localisation process, and none of them considers the vehicle's control and decision making in the loop. The focus of the next chapters will therefore be on active techniques, starting from navigation relative to an underwater structure, to a full deliberative system addressing active localisation.

3.6 Publications related to the Chapter

3.6.1 Journals

- Y. Petillot, **F. Maurelli**, N. Valeyrie, A. Mallios, P. Ridao, J. Aulinas, J. Salvi. Acoustic-based techniques for AUV localisation; *Journal of Engineering for Maritime Environment*, 224(4), 293-307, 2010

3.6.2 International Conferences & Workshops

- **F. Maurelli**, T. Larkworthy, D.M. Lane, G.C. Karras, C.P. Bechlioulis and K.J. Kyriakopoulos,; Pose-based and Velocity-based Approaches to Autonomous Inspection of Subsea Structures; *IEEE/MTS Oceans'13*, San Diego, United States; September 2013

- **F. Maurelli**, A. Mallios, D. Ribas, P. Ridao, Y. Petillot; Particle Filter Based AUV Localization using Imaging Sonar; *IFAC MCMC 2009*, Guaruja (SP), Brazil; September 2009

- **F. Maurelli**, S. Krupiński, A. Mallios, Y. Petillot, P. Ridao; Sonar-based AUV localization using an improved particle filter algorithm; *OCEANS 2009 IEEE*; Bremen, Germany; May 2009

- **F. Maurelli**, S. Krupiński, P. Sotiropoulos, A. Mallios, R. Haraksim; Lokiĝo kaj hejmeniro de memrega subakva roboto, KAEST 2008; Dobrichovice, Czech Republic; November 2008

- **F. Maurelli**, S. Krupinski, Y. Petillot, J. Salvi; A particle filter approach for AUV localization; *OCEANS 2008 MTS/IEEE*; Quebec City, Canada; September 2008.

Chapter 4

Localisation with respect to a structure

4.1 Introduction

In the past sections the problem of global localisation was analysed, i.e. the ability for the vehicle to understand its position in the general global coordinates, analysing several techniques with clear proposals for improved performances. Sometimes, however, the robot does not need to use this information, but it would rather uses information about its relative position to a specific structure. While manoeuvring around different types of underwater constructions - being them natural or man-made, the essential operational information shifts from the global frame to the relative one. According to the type of structure, the relative localisation problem can be defined in several ways. For the aim of this work, it is defined as finding a couple $< d, \theta >$, where d represents the distance from the robot to the structure, and θ represents the angle of orientation of the vehicle, with respect to the underwater structure. It is to be noted that, according to the chosen approach, d can represent a distance vector, with distances at different angles, and not just a single value. Linked to these parameters, the robot can enter into different behaviours, according the the chosen technique. In the following sections sonar information extraction is discussed and three different approaches are analysed and compared.

4.2 Structure Detection and Pose Estimation

This section analyses the relative pose estimation of the vehicle from the sonar image. The sonar mounted on $Nessie^{AUV}$ is a Tritech Gemini 720i Multi-beam Imaging Sonar, with a field of view of 120 degrees with a variable range extending between 0.2 and 120 meters. In the raw image data, the field of view is arranged in to 256 beams, while a maximum of 25 meters range at a scale of 120 pixels per meter in to

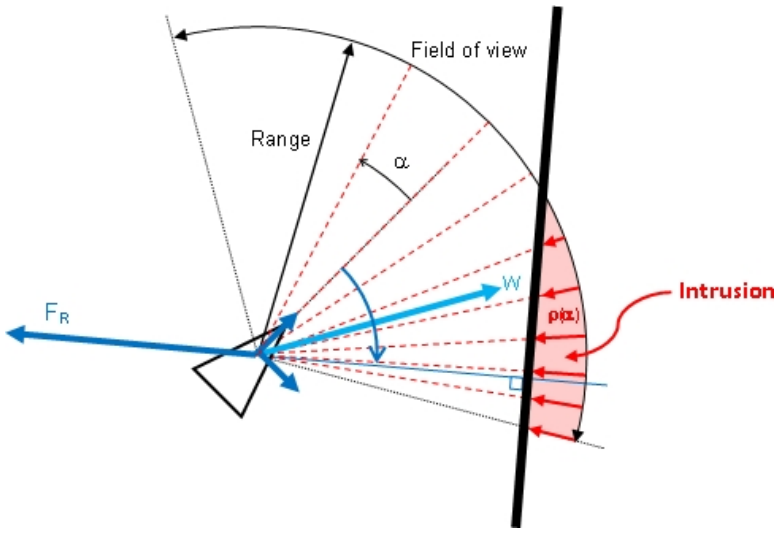


Figure 4.1: DVZ-Intrusion and reaction forces

the vertical bins, with a chosen frame rate of $2Hz$.

An empirically defined threshold is applied to the raw image to identify pixels corresponding to a strong reflection. The lower range sonar readings are ignored in the subsequent processing since they are a result of the AUV body structure's reflections. The object to inspect is assumed to be the closest object of reflection, as a result only the first bin close to the sonar is considered. Three approaches were analysed: based on RANSAC, on least square algorithm and on Hough transform. Considerations about the results are given in section 4.4.

4.3 Control Approaches

Three different approaches were analysed: the Deformable Virtual Zone, an *ad-hoc* Velocity Control, and Pose Request.

4.3.1 Deformable Virtual Zone

The first algorithm, based on a reflex behaviour reaction, uses the Deformable Virtual Zone (DVZ) concept, in which a robot kinematic dependent risk zone is located surrounding the robot. Deformation of this zone is due to the intrusion of proximity information. The system reaction is made in order to reform the risk zone to its nominal shape, implicitly moving away from obstacles [64]. The technique captures the orientation of the wall and the current situation of the AUV with respect to it, and drives the AUV reaction regarding its heading and distance to the wall, like explained in Figure 4.1.

4.3.2 Velocity Control

This technique, implemented by Karras *et al.* follows common practice in the relevant literature [54]. A kinematic control scheme is initially derived, considering the actuated velocities u, v, w, r as virtual control inputs (i.e., some appropriate desired velocities u_d, v_d, w_d, r_d are designed). Subsequently, the selected velocities are considered as reference velocities in the dynamic model and the actual control inputs X, Y, Z, N are designed.

The analysis will proceed in an Input to State Stability framework, i.e. the stability of the actuated degrees of freedom is first studied, assuming that p, q are absolutely bounded by some constants $\bar{p}$, $\bar{q}$, that is:

$$|p(t)| \leq \bar{p},\ |q(t)| \leq \bar{q},\ \forall t \geq 0 \tag{4.1}$$

and then prove that the overall closed loop system response does not violate the aforementioned bounds.

In this respect, the position and orientation errors are defined as $e_x = x - x_d$, $e_z = z - z_d$ and $e_\psi = \psi - \psi_d$. Notice, however, that a position error in y-axis is not defined since: i) constant velocity $\dot{y}_d$ in this axis is required and ii) an accurate estimate of y is almost impossible in the absence of absolute position measurements. To proceed, the following kinematic controller is chosen:

$$\begin{gathered}\begin{bmatrix} u_d \\ v_d \\ w_d \end{bmatrix} = \mathbf{J}_1^{-1}(\eta_2) \begin{bmatrix} -k_x e_x \\ \dot{y}_d \\ -k_z e_z \end{bmatrix} \\ r_d = -2k_\psi \frac{c\theta}{c\phi} e_\psi\end{gathered} \tag{4.2}$$

with k_x, k_z, $k_\psi > 0$ and the control inputs are designed as follows:

(4.5)

$$\begin{bmatrix} X \\ Y \\ Z \\ N \end{bmatrix} = \bar{\mathbf{M}} \begin{bmatrix} \dot{u}_d \\ \dot{v}_d \\ \dot{w}_d \\ 0 \\ 0 \\ \dot{r}_d \end{bmatrix} + \left(\bar{\mathbf{C}}(\mathbf{v}) + \bar{\mathbf{D}}(\mathbf{v})\right) \begin{bmatrix} u_d \\ v_d \\ w_d \\ 0 \\ 0 \\ r_d \end{bmatrix} + \bar{\mathbf{G}}(\eta)$$

where $\bar{\mathbf{M}}$, $\bar{\mathbf{C}}(\mathbf{v})$, $\bar{\mathbf{D}}(\mathbf{v})$ and $\bar{\mathbf{G}}(\eta)$ involve the rows of the corresponding dynamic model matrices, concerning only the actuated degrees of freedom (i.e., u, v, w, r). Finally, k_u, k_v, k_w, k_r are positive control gains and $e_u = u - u_d$, $e_v = v - v_d$, $e_w = w - w_d$, $e_r = r - r_d$ denote the velocity errors.

Considering an underwater vehicle and the control scheme in eq. 4.2, there exist positive control gains k_x, k_z, k_ψ, k_u, k_v, k_w, k_r such that the proposed control scheme solves the Structure Inspection task, despite the presence of modelling uncertainties and external disturbances.

4.3.3 Pose Request

This section presents a control approach based on pose request, rather than on force/velocity. It is useful in cases when the low-level control of the vehicle is not known, or the vehicle model is not available. Therefore it is based on giving pose requests to the pilot system, which will then translate those into force requests. Three cases are considered:

- **angular error**: if the angular error $|\theta_{err}|$ between the desired orientation and the current orientation is greater than a threshold θ_t, the vehicle only requests an angular adjustment. It does not move at the same time, because an angular error might be caused by obstacles at the side of the field of view, thus requiring an immediate rotation, in order to avoid the obstacle and continue to perform the inspection.

- **distance error**: if the vehicle is too close to the structure it is inspecting, the safest behaviour is to ask only for the requested distance from the wall, and adjust any angular error. Being too close to the structure, the field of view is very limited and the safest behaviour is to return to a stable situation, i.e. at the desired distance and desired angle, before re-engage in the dynamic inspection.

- **normal situation**: this happens when $|\theta_{err}| < \theta_t$ and $|dist_{real} - dist_{desired}| < t_{dist}$. In this case, the waypoint is computed and requested in order to complete the inspection.

This technique doesn't require for the waypoints to be reached by the vehicle. They can have arbitrary distance, as new waypoints are issued at every sonar frame. Compared to the previous approach, it can handle heterogeneous structures, and it doesn't require any knowledge of the vehicle model. On the other hand, velocity-based control

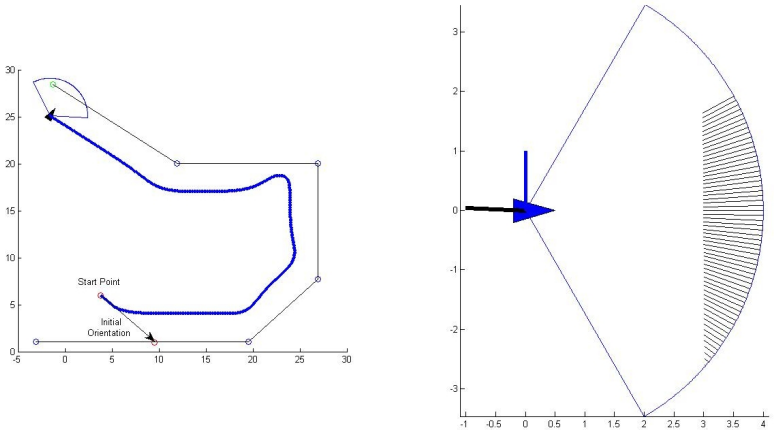


Figure 4.2: (left) Trajectory tracking; (right) sonar simulation and field of view (Gemini sonar)

is much smoother in defined situations, where lines are the predominant feature of the structure.

4.4 Experimental Results

Simulation and in-water trials successfully demonstrate the validity for all techniques, with some advantages for the pose-based one.

4.4.1 DVZ

For the DVZ technique, two different types of sonar were evaluated: the Tritech Micron Scanning Imaging Sonar, mounted on Nessie IV AUV and the Tritech Gemini 720i Multibeam Imaging Sonar, mounted on Nessie V AUV. The first one has a field of view of 360 deg, but, being mechanically scanning, is relatively slow in computing the image, compared to the second. The Gemini sonar is a multibeam sonar, with 256 beams, but covering only 120 deg. Figure 4.2 shows simulated results, highlighting the trajectory of the vehicle. Figure 4.3 reports more analytical data of the distance between the vehicle and the structure and the velocity on each of the vehicle axis.

Experimental results with the robot Nessie IV happened at the Somerton Diving Pool (1.7.4) and at the QinetiQ Ocean Basin Tank (1.7.5). Experimental results with the robot Nessie V happened at the OSL Wave Tank (1.7.2) and at the CMRE Waterfront (1.7.6). Unfortunately, the performances in open-water were not as good as expected. The algorithm showed weaknesses in presence of currents and waves, which were not estimated in the model.

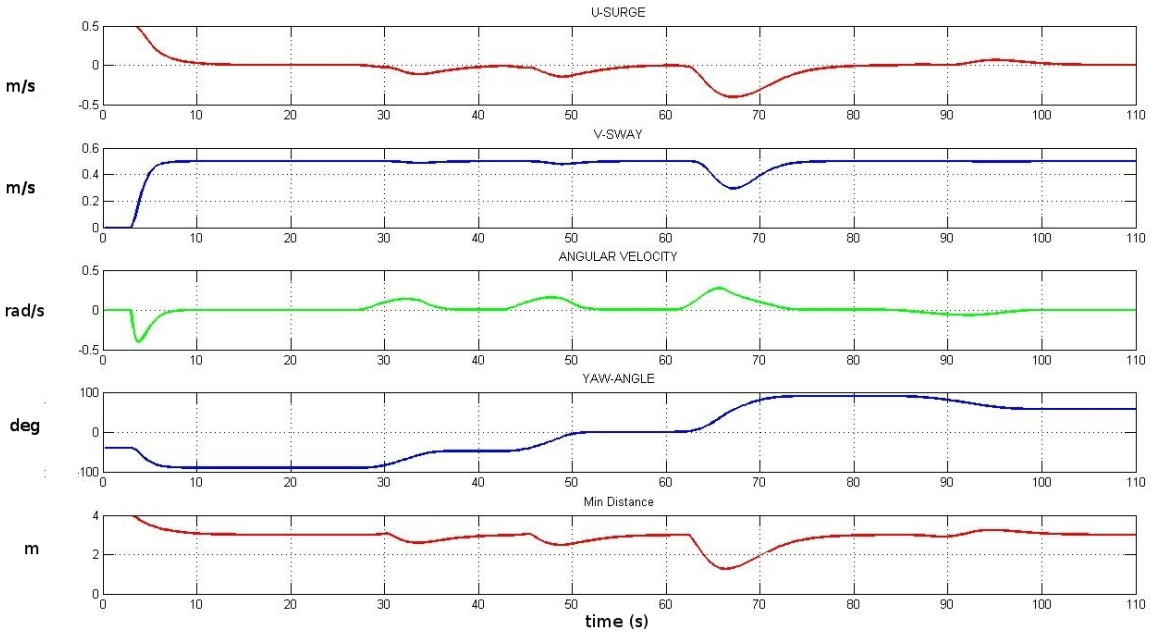


Figure 4.3: Temporal evolution of angular velocity and velocities on surge and sway

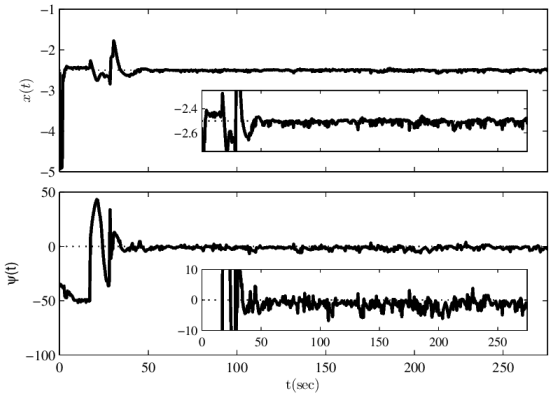


Figure 4.4: No disturbances: The distance and orientation with respect to the wall along with the desired values.

4.4.2 Velocity Control

The velocity-based approach was evaluated at the Wave Tank. Two sets of experiments were considered: the first one without disturbances, and the second with disturbances (i.e. medium waves). In both cases the vehicle responded very well, as highlighted in Figure 4.4 and Figure 4.5. Figure 4.6 shows the vehicle coping with waves and disturbances.

4.4.3 Pose Request

The pose-based approach was evaluated at the Wave Tank and at the CMRE Waterfront, during SAUC-E 2010. The condition at the CMRE waterfront were not

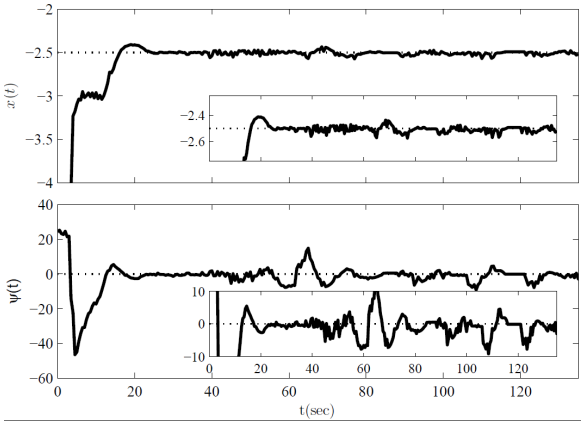


Figure 4.5: In the presence of disturbances (medium height waves): The distance and orientation with respect to the wall along with the desired values.

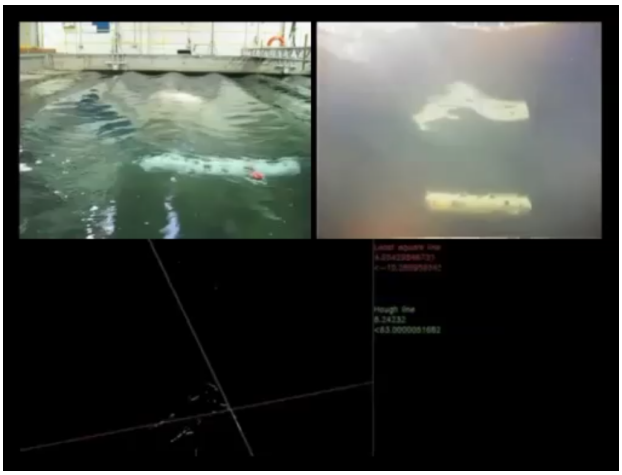


Figure 4.6: The vehicle autonomously surveying the wall in presence of waves. The sonar data and first analysis is shown at the bottom of the Figure.

as controlled as in the Wave Tank, as current sea conditions were depending from several conditions: a fresh water stream, on specific times of the day, waves caused by weather conditions, waves caused by passing ships. The algorithm was tested in various conditions, at different times of the day, and the performances were always very satisfactory. Analytical results show that the error is always bound in between 10 cm for patches of clear wall, even in case of disturbances. Close to corners, the error increases, due to the limited field of view of the sensor. However, as soon as the turn is completed, the error is again bounded. Figure 4.7 shows the vehicle surveying two walls. The algorithm is general enough to handle corners and not linear surfaces.

4.4.4 Considerations

Analysing the results of the different trials, it is possible to see that a RANSAC approach is more robust in the case of clear wall, whilst a least square approach allows the vehicle to successfully overcome obstacles and corners. The best structure identification algorithm therefore can be dependent on the type of structure being inspected. RANSAC parameters are also very problem-dependent and the output is the one best line fit, discarding other points. In complex structures, there is the risk that only a portion of the structure is considered, i.e. the best line approximation in a subset of the image. In cases of irregular shapes and corners, a RANSAC approach is not able to cope with those, whilst a least square solution considers all the sonar return and is therefore able to take into account a shape change, as soon as it is visible, from the very beginning. Therefore, although RANSAC is to be preferred in case of wall following, with no corners, a least square approach is more general and able to address a wider range of environments, with no specific shape definition.

A velocity-based approach provides smoother trajectories, mainly in presence of clear walls, and allows to define the desired velocity of the vehicle. A pose-based approach, on the other hand, is more general, and depends on geometric calculation only, leaving all aspects related to the vehicle itself out of the loop. In this case, it is not possible to directly control the velocity, but the pose request does influence the velocity, if the internal control of the vehicle is based on a PID.

4.5 Conclusions

This Chapter has presented several techniques to solve the problem of robot navigation around structures. It is a different approach than the ones presented in the previous chapters, as the vehicle's control is in the loop. It is however a reactive

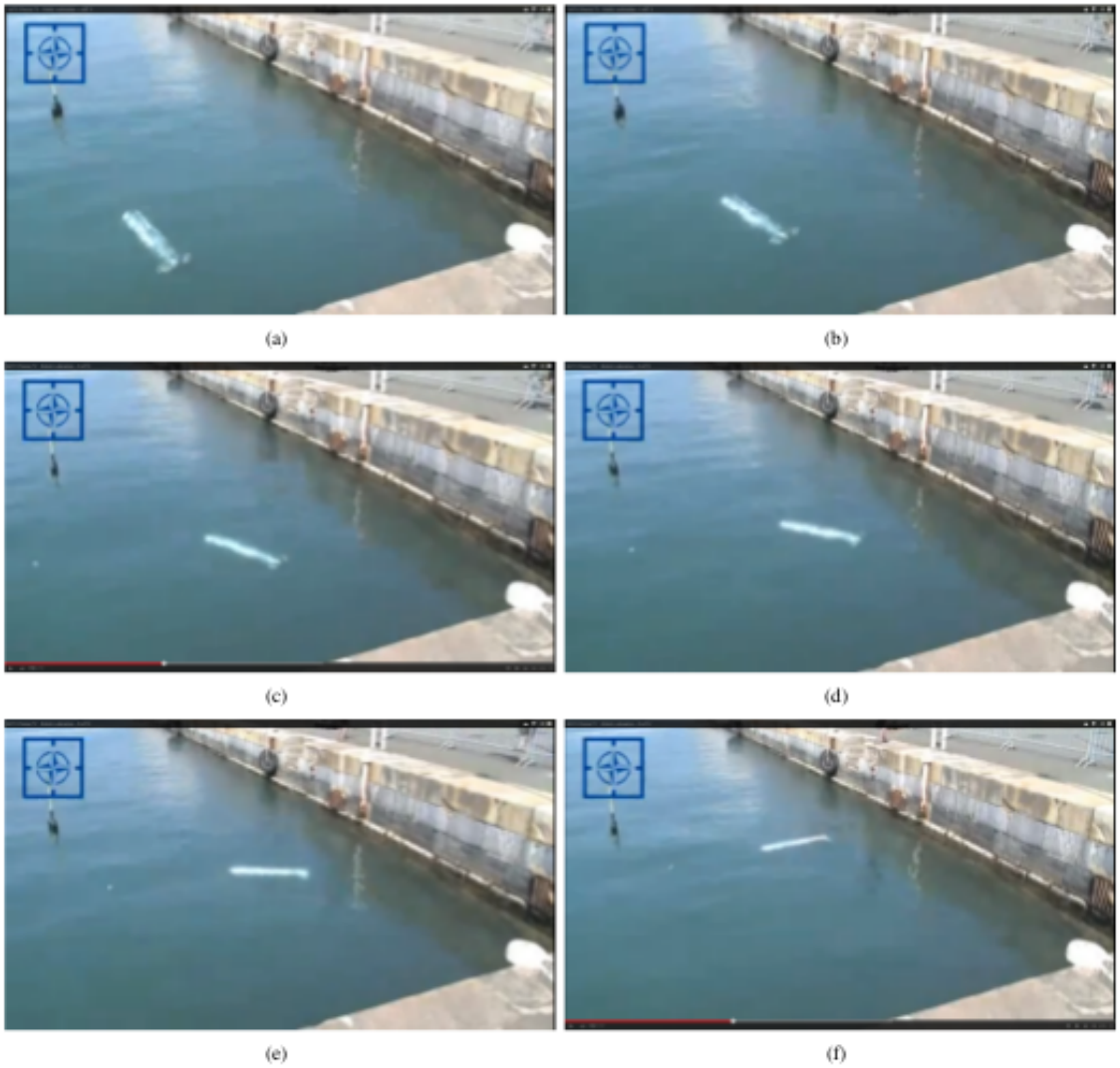


Figure 4.7: Wall Inspection at NATO CMRE Waterfront. The pose-based approach is able to survey any surface, and overcome angles, as well as unstructured environment. In the Figure, the vehicle is able to turn itself using a very simple and generic approach, not dependent on the specific scenario.

behaviour for navigation and localisation, whilst in some circumstances a deliberative layer of active localisation is needed. In the cases analysed in this chapter, the vehicle responds to the sensor data directly, whilst a further step forward in the active navigation topic is a more complex vehicle reasoning, able to support the localisation process, which is not dependent on the current sensor frame, and which takes into account the full probability density function of the estimate of the vehicle's state. The next Chapters will therefore analyse and propose a solution to the active localisation problem.

4.6 Publications related to the Chapter

4.6.1 Journals

- **F. Maurelli**, J. Cartwright, N. Johnson, Y. Petillot; Nessie IV Autonomous Underwater Vehicle wins the SAUC-E Competition; *Robótica*; 82,10:15, 2011;

4.6.2 International Conferences & Workshops

- R. Baxter, J. Cartwright, J. Clay, O. Clert, B. Davis, J. Lopez, **F. Maurelli**, Y. Petillot, P. Patron, N. Valeyrie; Nessie V: a new torpedo-shaped hover-capable autonomous underwater vehicle for survey, inspection and intervention; *AUVSI's Unmanned Systems North America*, Denver CO, USA; August 2010

- **F. Maurelli**, Y. Petillot, A. Mallios, P. Ridao; Benchmarking Autonomous Underwater Vehicles: the example of the SAUC-E competition ; *Workshop on Benchmarking Intelligent (Multi-)Robot Systems, 19th European Conference on Artificial Intelligence (ECAI'10)*, Lisbon, Portugal; August 2010

- **F. Maurelli**, J. Cartwright, N. Johnson, Y. Petillot; Nessie IV Autonomous Underwater Vehicle wins the SAUC-E Competition; *IEEE Conference on Mobile Robots and Competitions*, Leiria, Portugal; March 2010 - **Award Winner** as best scientific paper!

- S. Krupinski, **F. Maurelli**, A. Mallios, P. Sotiropoulos, R. Haraksim, T. Palmer; Towards AUV docking on subsea structures; *OCEANS 2009 IEEE*; Bremen, Germany; May 2009

- S. Krupinski, **F. Maurelli**; Localisation and guidance in the AUV docking problem; *Workshop in visual guidance systems for small autonomous aerial vehicles, IEEE/RSJ IROS 2008*; Nice, France; September 2008

- S. Krupinski, **F. Maurelli**, G. Grenon, Y. Petillot; Investigation of autonomous docking strategies for robotic operation on intervention panels; *OCEANS 2008 MTS/IEEE*; Quebec City, Canada; September 2008

Chapter 5

Active techniques for AUV localisation

5.1 Introduction

Chapters 2 and 3 have showed several approaches to AUV localisation, with the development of novel techniques, comparing different methods and addressing a fully known map as well as a partially known one. In all the presented approaches, however, the robot was "simply" processing the data coming from the sensors (usually motion estimation and measures of the environment). There was no robot motion control involved. Chapter 4 presented localisation with respect to a structure and algorithms to perform an inspection. However the motion was not related to any global localisation, but only to relative navigation.

This chapter will now focus on *active* techniques, with an emphasis on intelligent decision making, for localisation. By the word *active*, the author means *with the vehicle control in the loop*, so with the robot *actively* choosing an action, in order to facilitate self-localisation. Adding the control in the loop represents a way to improve the robustness and the efficiency of the process. In certain cases, it might also be the only way to successfully solve the problem of self-localisation. The key difference is that an active selection of the best set of actions to be executed is performed, in order to reduce the uncertainty, rather than just passively evaluating data from the sensors.

The adjective *active* is therefore referred to the robot who *actively* chooses a set of actions. It is to be noticed that sometimes the literature referred to *active localisation* in relations to localisation with *active features* (e.g. active beacons, lights), see for example in [36, 35, 93]. The two should not be confused, as they mean completely different approaches.

This chapter will analyse the current state of the art, in the robotic field more in general, as literature specific for the underwater domain in active localisation is

very limited. Additionally, some of the techniques presented for land robotics, may be applied in the underwater domain, with appropriate modifications.

This Chapter is organised grouping the different available techniques:

- section 5.2 will present the active selection of landmarks in EKF;
- section 5.3 will present the use of Multiple-Hypothesis Kalman Filters in the context of active localisation;
- section 5.4 will present the notion of *entropy* in the localisation probability density function, linking active localisation to the minimisation of the entropy;
- section 5.5 will present an approach of action selection to facilitate the localisation process;
- section 5.6 will analyse a typical approach for active localisation in underwater robotics, in a scenario where the localisation process is aided by an acoustic beacon;
- section 5.7 will present the link between active localisation and path planning, showing how the two can be linked;
- section 5.8 will present multi-robot approaches to active localisation;
- section 5.9 will then present other approaches, who are not classifiable in any of the other categories.

At the end of the Chapter there is an overall critical analysis (section 5.10), showing the shortfalls of current techniques, giving a hint of the direction of the next chapter, which represents one of the main contributions of this thesis.

5.2 Active Landmark Choice

The first presented approach deals with *active landmark choice*. The vehicle's navigation system is implemented using an Extended Kalman Filter (EKF, see section 2.4). As discussed, Extended Kalman Filter is a landmark-based localisation system. The algorithm can be diveded into two phases: *Prediction* ad *Update*. Observation of the landmark triggers an update in the filter, to correct the predicted state estimation. Instead of passively searching for landmarks in the measurement, this approach actively influences the vehicle in order to actively look for landmarks, in order to reduce the uncertainty. It is therefore applied whenever there is a predicted location and an associated uncertainty. From the position uncertainty ellipse, the visible landmarks are selected, and the best landmark is then chosen.

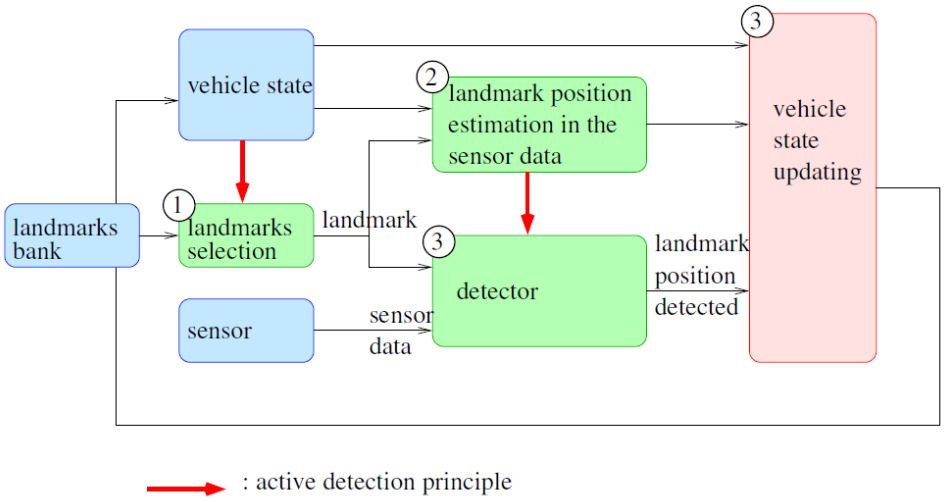


Figure 5.1: Localisation system proposed by Tessier [105], based on active landmark selection.

Arsenio & Ribeiro applied this technique to mobile robots in a controlled indoor environment [7]. This approach helps tracking the position of the robot, especially in presence of landmarks which are not easily observable (in the specific case: glass wall, with a laser scanner). The system therefore chooses to look at more observable landmarks. If the uncertainty grows, a specific module is called for global localisation. There is however no specific strategy to discriminate between different possible locations of the vehicle.

This idea is also proposed by Tessier *et al.*, in [105], where the landmarks are actively selected, by a supervisor module. The proposed strategy of active landmark detection optimises a combination of tools - a landmarks bank, sensors and detectors - by introducing the notion of perceptive triplets. Figure 5.1 presents the localisation approach and the active detection principle. This approach is very useful and can be ported in the underwater world, in the case distinctive landmarks can be detected. Similar to the previously described approach, it does not address the initial pose estimation and the disambiguation between multiple possible locations of the vehicle.

Olson present techniques to optimally select landmarks for performing mobile robot localisation by matching terrain maps [86]. The method is based upon a maximum-likelihood robot localisation algorithm that efficiently searches the space of possible robot positions. A sensor error model is used in order to estimate a probability distribution over the terrain expected to be seen from the current robot position. The estimated distribution is compared to a previously generated map of the terrain and the optimal landmark is selected by minimising the predicted uncertainty in the localisation. In order to predict the uncertainty obtained by localisation using various landmarks, the proposed method constructs a probabilistic representation of the

terrain expected to be sensed at any position in the global map. Treating the patches of this *probability map* of the terrain as a local map allows the uncertainty expected by sensing the terrain patch to be estimated using the surface fitting techniques. This results in a rocky terrain were quite promising. Similarly to the previous paper in this area, this work can be applied in the underwater domain with the necessary adaptation in cases where distinctive landmarks can be selected. It suffers however of the typical limitations already described in this section.

5.3 Multiple-Hypothesis Kalman Filter

Remaining in the Kalman Filter domain, a possibility to consider multiple possible locations for the vehicle is the use of a *multiple-hypothesis Kalman filter*. This is based on multiple Kalman filters running in parallel, each carrying a possible location, with related uncertainty. Each hypothesis is represented by a pose estimate $\hat{x}_i = (\hat{x}, \hat{y}, \hat{\theta})_i^T$, with an associated covariance matrix, Σ_i, and information about the probability of the hypothesis being the correct one $P(H_i)$. The consideration of multiple hypothesis is extremely important when the problem is not only position tracking, but also global localisation. Figure 5.2 shows an example where multiple hypothesis are essential. The robot is in a simple environment with one room and four door. The perception system recognises a door. Therefore there are eight possible location, or *hypothesis* for the robot. In this framework, the *active* approach consists in the determination of the best move that maximise the expected number of new features observed.

Jensfelt & Kristensen proposed an approach based on multiple hypothesis Kalman filter [47]. They used a topological map to represent the environment. The decision on the robot move is determined by the maximisation of the expected number of new features observed in the next possible moves. Starting from N possible hypothesis, the

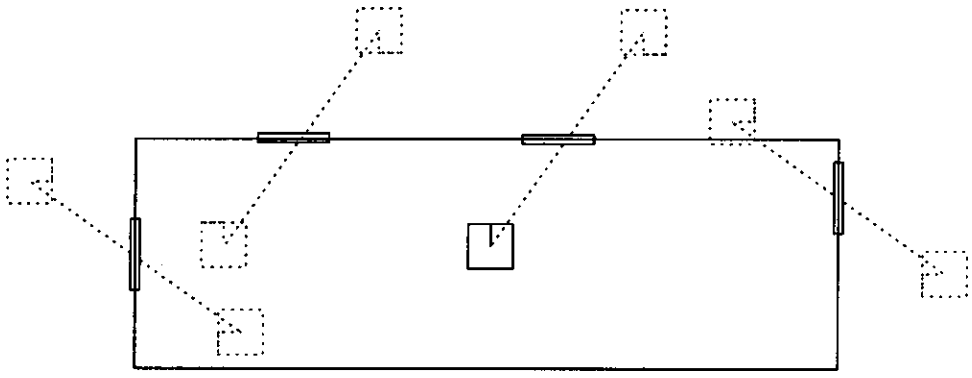


Figure 5.2: The need for multiple pose hypothesis shown in a simple environment with only one room with four doors. The robot can see a door, thus there are eight possible locations (or *hypothesis*).

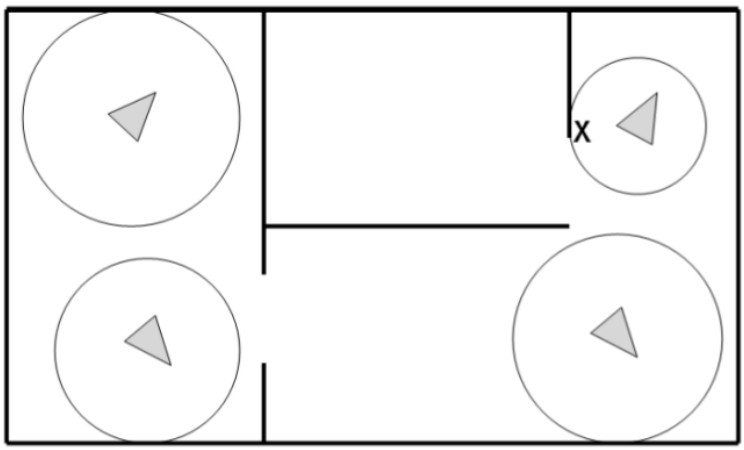


Figure 5.3: The selection of the waypoint in [34], selecting a point close to the nearest possible obstacle

most probable is considered. A search in the topological graph is performed from the current location of this hypothesis in order to find the one of the neighbouring nodes not previously visited and providing the largest number of features. This node is then selected as the next node to go to. This approach works well when it is possible to identify a clear set of features. The exploration is driven by some heuristics, including the avoidance of visiting the same location twice. This is an important optimisation, as visiting the same location twice does not provide any new information.

The work of Gasparri *et al.* is also based on multiple hypothesis Kalman filter [34]. The algorithm relies on two steps: hypothesis generation and safe planning and tracking technique. The first step exploits a particle filter to find out the most likely hypotheses with the assumption of stillness of the robot. The second step plans safe trajectories to reduce the remaining ambiguities using an extended Kalman filter for each hypothesis when the robot is moving. Figure 5.3 shows the active selection of a waypoint in the second step of the algorithm. It is close to the nearest possible obstacle, to allow a safe navigation. A good strategy of this approach is the use of two steps, one to identify a finite number of poses and another one to disambiguate among the poses. The simulation results are very promising. It is however unclear how the algorithm can avoid deadlock situations, which can arise in some environments where the selection of a single waypoint might not be enough. Surely the algorithm can be applied sequentially, which would help in many cases, though not solving the possible deadlock in the disambiguation problem.

5.4 Entropy minimisation

When the robot localisation is performed with Markov-based techniques (for example in section 3.2), a different approach to *active* localisation can be chosen. Considering a location l, defined as $l = (x, y, \theta)$. The distribution, denoted by $Bel(l)$, expresses the robot's subjective belief for being at l. $Bel(l)$ is updated in two cases. In the first one, the update is triggered by a robot motion (or action more in general). Modelling the motion in probability terms, $P_a(l/l^1)$ represents the probability of being at location l, after executing the action a from position l^1. As explained by Fox *et al.* in [33], the believe is then updated using the following formula:

$$Bel(l) \leftarrow \int P_a(l/l^1)Bel(l^1)dl^1 \tag{5.1}$$

The second case when the belief is updated happens when the robot sensors provide a measurement. Considering s as a sensor reading, and $P(s/l)$ the likelihood of perceiving s at l, $Bel(l)$ is updated using the following formula:

$$Bel(l) \leftarrow \frac{P(s/l)Bel(l)}{P(s)} \tag{5.2}$$

To eliminate uncertainty in the position estimate $Bel(l)$, the robot must choose actions which help it distinguish different locations. The entropy of the belief measures the uncertainty in the robot position and is obtained by the following formula:

$$H = -\int Bel(l)log(Bel(l))dl \tag{5.3}$$

If $H = 0$, $Bel(l)$ is centred on a single position. In this framework actions are selected in order to minimise the expected future entropy. Considering $Bel_{a,s}(l)$ the belief after executing the action a from location l and sensing s, the expected entropy can be represented by:

$$E_{a,s}[H] = -\int Bel_{a,s}(l)log(Bel_{a,s}(l))dl \tag{5.4}$$

The expression expected entropy after executing action a is obtained by integrating over all possible sensor values s, weighted by their likelihood, and by applying the

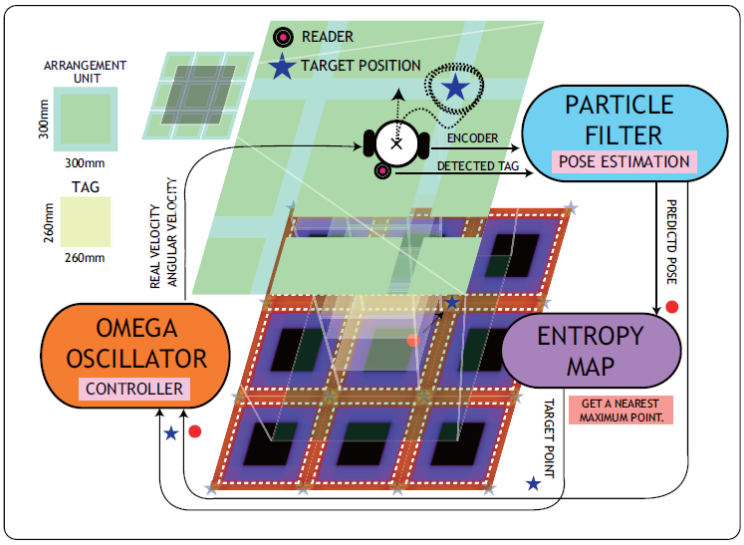


Figure 5.4: The algorithm proposed by Kodaka [57], based on a pre-calculated entropy map.

update rule, defined in Eq. 5.2:

$$\begin{aligned} E_a[H] &= \int E_{a,s} p(s) ds \\ &= -\int\int Bel_{a,s}(l) log(Bel_{a,s}(l)) p(s) dl ds \\ &= -\int\int P(s/l) Bel_a(l) log \frac{P(s/l) Bel_a(l)}{P(s)} dl ds \end{aligned} \tag{5.5}$$

Solving the problem of *active* localisation therefore means to minimise $E_a[H]$.

A very important work using this technique is made by Burgard *et al.* [13] and Fox *et al.*[33]. In their work they selected actions by maximising the weighted sum of the expected decrease in uncertainty (entropy) and the costs of moving to the target point. Target points are specified relative to the current robot position and can represent an arbitrary point in the space. Path planning is not involved in the active localisation module. The result of the algorithm is only a single point to be reached by the robot. Position probability grids are used to estimate the vehicle position.

Kodaka *et al.* proposed an approach for mobile robots based on an entropy map [57]. RFID tags have been placed into the environment with the entropy map precalculated based on the arrangement of the tags. After a pose prediction using particle filtering, the robot is attracted to the target using a dynamic model, the fundamental unit of which is rotation-based angular velocity. Figure 5.4 shows the steps of the algorithms and the relations with the environment. This solution is too specific to be applied in the underwater domain with similar scales. Using active beacons, with

greater distances due to the physical nature of the sensors, it is possible to design a similar solution. However, the different constraints in the vehicles would suggest an alternative solution which would suit better the underwater world.

Mariottini & Roumeliotis presented an active vision-based localisation technique in a large-scale image map, represented as a vocabulary tree [70]. They adopted a sequential Bayesian approach in order to eliminate the localisation ambiguity by exploiting additional camera measurements over an extended time horizon, while navigating towards a target image, and along the least-ambiguous (i.e., low entropy) visual path.

Kümmerle *et al.* use this approach involving active sensing, i.e. the possibility for the robot to decide where to point the sensor. Active sensing represents a subset of the full active localisation problem, where the possible actions the robot can undertake are limited to the pointing of the sensor. They use particle filters for the vehicle localisation (section 3.2). They cluster the particles into groups and calculate the total expected entropy for the particle filter by a weighted average of the expected entropy for each cluster/group [61].

5.5 Selection of best action

This section presents a selection of approaches which are based on the selection of the single best action for the robot to undertake, in order to localise itself. The method to discriminate between the different actions can be different, but they all have the same framework in common: given a set of n possible actions $A = a_1, a_2, a_3, ..., a_n$, the algorithm select the action a_i, which maximises the following formula:

$$i = \underset{i}{argmax}(reward_{a_i} - cost_{a_i}) \tag{5.6}$$

The work carried on by Fairfield & Wettergreen represents an important contribution using this approach, also because it is one of the few examples in the underwater domain [30]. It uses active localisation on top of the map previously constructed by a SLAM approach. The set of possible actions are represented by the heading of the vehicle for the following $30m$. The action is selected in order to choose the most discriminative one. The vehicle state is represented with a particle filter, and only a subset of particles are used to evaluate the best action. In many cases however a single action - in this case: setting the heading for the following $30m$ - is not enough to discriminate between multiple hypotheses.

Solberg *et al.* propose an approach based on the active movement of an electric field emitter [102]. Their approach is based on electric fields for vehicle navigation,

a biology-inspired concept. The vehicle state is estimated with a particle filter and the chosen control option minimises the expected variance of the particles at the next iteration.

Seifzadeh *et al.* propose some modification to the standard Monte Carlo localisation, in order to solve the kidnapped robot problem and to initialise the particles in a more efficient way [98]. In the description of the active approach, the robot chooses an action which represents the best trade-off between cost and gain. Again, similar to other approaches discussed above, choosing only one action does not guarantee any result in complex environment, and thus can only be used in very specific cases.

Chuho *et al.* developed an active-semantic localisation method [122]. A Bayesian model for robot localisation has been applied, incorporating also spatial contexts among objects, which were described using symbols. The robot action selection is based on a greedy approach. Only the best action is considered.

Murtra *et al.* also presents an approach based on the selection of the best action to execute [80]. It is based on a rational criteria to select the action that minimises the expected number of remaining position hypotheses, using a Particle Filter.

5.6 Beacon-aided localisation

Some approaches in active localisation can be very specific and tailored to a specific robot configuration or with specific environmental constraints. This is the case of the approach of Olson *et al.*, who use active beacons deployed in the environment, in order to help the localisation process [87]. The use of active beacons (i.e. acoustic emitters) is quite common in underwater robotics. Using this approach, there are two standard solutions to the localisation problem. The *active* approach to localisation therefore aims to disambiguate between these two solutions, with a specific path to be followed. Figure 5.5 shows the exploration gradient with two possible beacon locations. The best disambiguating motion is a function of the AUV's location. The vehicle maximises the difference between the range measurements by travelling along the arrows. The length of the arrows indicates how rapidly the difference in range changes. This approach is capable of performing localisation without relying on carefully surveyed beacon locations. The ability to localise a beacon is tightly coupled to the path travelled by the AUV. The robot's path should be therefore chosen to optimally resolve ambiguous data. Although simulation results were very promising, this approach cannot be easily generalised or adapted if beacons are not present in the environment.

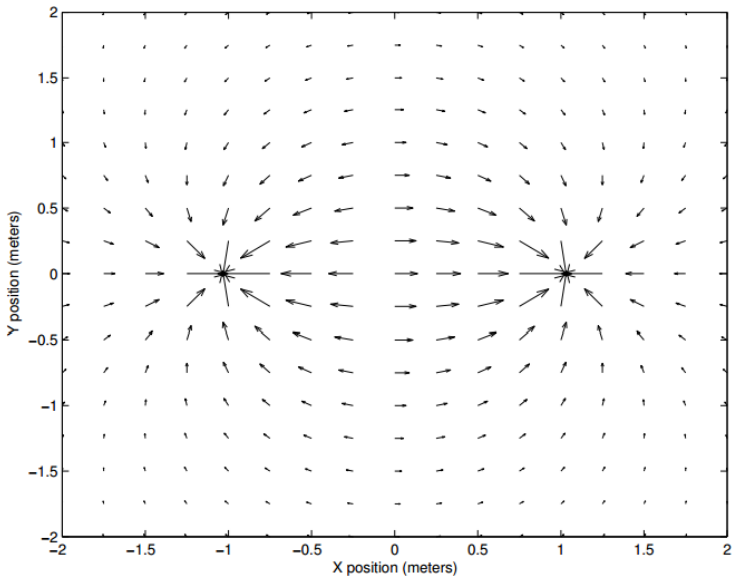


Figure 5.5: Exploration gradient with the beacon location at (-1;0) or (1;0). The best disambiguating motion is a function of the AUV's location.

5.7 Path planning and active localisation

Active localisation can also be seen not just as an isolated problem, but linked to the overall robotic system, which has tasks to perform. It might not always possible to focus on localisation neglecting any other goal of the robot. On the other hand, the opposite is almost always impossible: the knowledge of the robot state is very often - if not always - a needed condition. A way to address this area is to perform active actions, which would contribute both to the localisation process and to the end goal. In trajectory planning, for example, a localisation-aware trajectory would be not necessarily the shortest one to the goal point, but the one which would allow the robot to see features and arrive to the goal point with a reduced uncertainty. The work of Bauer is in this area and presents an approach to support the data acquisition for the localisation process of an autonomous robot by well-aimed manoeuvres [10]. The task of localisation is linked with a specific goal to be reached. In the case of path planning, the proposed approach mediates among the different tasks: localisation and user defined mission. This mediation is performed by analysing the estimated benefits and cost of each task and selecting therefore the optimal manoeuvre. The work was tested in simulation with line features as the landmark types for the robot to localise itself. Important assumptions for this work are the knowledge of the start position and the limitations to line features.

5.8 Multirobot active localisation

In some cases the problem of localisation and active localisation can be addressed in a multi-robot scenario, showing the benefits of active approaches versus passive ones. Bhuvanagiri & Krishna designed a system to guide several robots who are in ambiguity of their states to locations where as many of them can get rid of their ambiguities by localising to a unique hypothesis state [12]. It presents a unified probabilistic framework that takes into account the role of measurements between robots as well as the measurement made on the local map structure in deciding the best locations to move. The robots choose to move towards those locations where the probability of localising itself to a unique hypothesis is maximum. This work shows the advantages of a multi-robot system in addressing problems such as state estimation. Davison & Kita demonstrated accurate localisation for an inspection team consisting of a robot with stereo active vision and its companion with an active lighting system [24]. In this case a single sensor can be used for measuring the position of known or unknown scene features, measuring the relative location of the two robots, and actually carrying out an inspection task. The active vision system is based on active landmark choice, described in Section 5.2

A close multi-vehicle collaboration is however out of the scope of this thesis.

5.9 Other approaches

In this section other specific approaches, which cannot be grouped in the categories described above, are presented. They often provide a customize solution for a specific problem, thus making their portability in other domains or in scenarios with different constraints difficult.

Antonelli *et al.* focus on the improvement of observability for relative localisation of AUVs [6]. The case of cooperation between two vehicles is analysed and numerical simulations have shown path configurations which avoid singularities. The proposed approach is purely mathematical dealing with system observability. A system is said to be observable if, for any possible sequence of state and control vectors, the current state can be determined in finite time using only the outputs. The approach evaluated valid paths which allow full rank observability matrix for the linearised system, i.e. all variables are fully observable. Those paths are however defined in advance, and specific to the situation. The behavioural control techniques described in the paper do not provide a general solution to the localisation problem, in cases where the robot needs to choose the best trajectory (or, in a wider sense, the best set of actions).

O'Kane & LaValle have analysed three robot configurations with limited sensing,

in order to analyse the possibility for a simple robot to localise itself in a polygonal environment [84]. A discretisation of the state space is applied. No uncertainty is considered, but all the possible states are in a finite set. In order to disambiguate, two random possible states are considered, and a list of actions is computed in order to arrive to a point when only one of the two states is admissible. This approach, justified with algorithmic proofs, successfully determines the robot pose. Among the limitations, the environment needs to be polygonal, no uncertainty is taken into account and the two possible poses among the set of all possible poses are chosen randomly, which leads to the possibility of having to apply recursively the algorithm $n-1$ times, with n being the number of possible initial states.

Dudek *et al.* presented a method for minimum distance traversal for localisation that works in polygonal environments without holes that they show to be NP-Hard [26]. A randomized version of the same method was presented in [92].

The work proposed by Kondo *et al.* on localisation around underwater structure is an example of linking the localisation with the planning system [58]. The localisation itself is based on particle filters (section 3.2) but the planning system considers the state of the filter in its planning. Once convergence is reached, it issues waypoints, in order to inspect the structure at a fixed distance. In the proposed system, the link among localisation and planning is not very strong. The planner waits for a stable navigation status, before moving the vehicle, which is very reasonable, but does not consider the possibility that a consistent status might be achieved only after a specific set of actions.

5.10 Critical Analysis

The previous sections presented several approaches for active localisation. Actively choosing the landmark to observe represents a good strategy for landmark-based localisation. However, it does not address the disambiguation among multiple possible locations. Multiple-Hypothesis Kalman Filters address the possibility of multiple possible locations, but this approach works well only when it is possible to identify a clear set of features. The concept of entropy minimisation is considered key from the author. Considering the localisation function as a probability distribution function, any active localisation technique directly or indirectly needs to minimise the expectation of the future entropy, when selecting the actions. However, a clear drawback is represented by the complex mathematical formulation, which is computational intensive, as also outlined by Fairfield & Wettergreen ([30]). Additionally, the solution provided by Burgard *et al.* [13] provides one map point relative to the robot which the robot should reach to minimise the entropy in the localisation distribution. However that

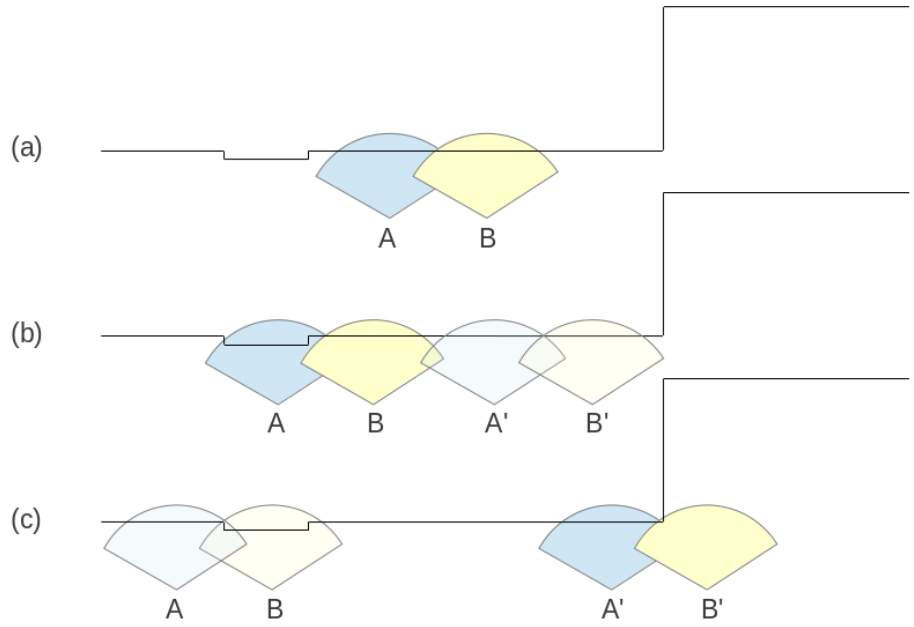


Figure 5.6: An example of why building a set of actions greedy concatenating the best single action is not a powerful solution. A and B represent two possible location of the robot. (a) initial situation; (b) the best single action is to move towards the left; (c) the best single action from the previous best single action is again a move towards the left. However, if the robot moves towards the right twice from the initial situation, it arrives to a much better location to discriminate among the two hypothesis.

point might not be accessible, and the path planning to reach that point is not explicit. The selection of the best action works well only in simple environments. In many cases there is the need of a series of action in order to correctly estimate the location in the map. Executing one action only does not give any guarantee to improve the localisation. A simple greedy approach of building up a new action on top of the previous best action is again not suitable, due to the possibility of local minima and local maxima. Figure 5.6 shows an example. Considering two possible states, A and B, the single best action is to move towards the left, as moving towards the right would not discriminate at all among the two possible location. If the system would build a set of action composing the best single actions, it would then go again towards the left. However, if the system chooses to go right from the beginning, it arrives to a much better location to discriminate among the two hypothesis. The work from Gasparri *et al.* [34] shows a proposal which is very similar to the selection of the best action, though the selection process is more elaborated, therefore having similar limitations. The selection of a point close the first possible obstacle was mainly justified because of safety. It actually helps the whole localisation process. The idea of looking for places which are different according to the different hypotheses is key. Based on

the literature analysis, a clear gap is identified. The proposed system needs to:

- be general enough, and not tight to any specific custom problem;
- be able to handle multiple possible locations (global localisation), and not just the current one (position tracking);
- consider the entropy expectation as an important information in the definition of the actions to be performed;
- consider multiple actions, as one action is not enough to properly address the problem;
- optimise the computational load

The following chapter will present and formalise the proposed system, taking into account the above-mentioned criteria.

Chapter 6

Novel approaches for AUV active localisation

6.1 Introduction

The previous chapter has showed several approaches to AUV active localisation. This Chapter will now focus on a novel deliberative active system which addresses the shortfalls of the current approaches. The proposed system aims to be general enough, not tight to a specific environment or sensor. Handling multiple possible locations (global localisation) needs to be one the features, and not just performing position tracking. Additionally, it needs to consider multiple actions, because - as outlined in the previous chapter - choosing the best one action is not enough. The proposed module is based on a Particle Filter approach, described in 3.2 and is able to return a set of actions in order to facilitate particle convergence.

This Chapter is organised as follow:

- section 6.2 will present the active localisation module, one of the main contributions of this thesis, with comparison with other techniques;
- section 6.3 will present the experimental results, both on simulated data and in field trials;

Finally, conclusions will be presented in 6.4

6.2 AUV Localisation Module

The proposed approach takes into consideration the analysis made in the previous section and addresses all the highlighted points. An architectural overview is firstly presented, while each module will be described in more detail in the following sections.

The integration with the vehicle architecture is a key, and it is reasonable therefore to assume that the system needs to continuously run the localisation module. In specific cases, there should be the possibility of actively taking the control of the vehicle in order to execute a plan aiming at reducing the uncertainty in the localisation. Therefore, the proposed system, outlined in Figure 6.1, is a two-layer localisation architecture. The first layer represents a continuously running *passive* localisation module, as described in Chapter 3, section 3.2. The second layer is an *active* module, i.e. with the control of the vehicle's motion in the loop. The first module is the one running by default: it receives the information from the vehicle's sensors (for example Doppler Velocity Log, compass, depth and altimeter sensors, sonar measures) and it combines them in order to estimate the vehicle state. When there is the need to actively localise, the second module starts and generates the best set of actions to be executed, in order to reduce the uncertainty. The decision to switch from passive localisation to active localisation needs however to be taken by the mission planner, based on the current goals and vehicle state estimation. Considering the full vehicle architecture, a single module should not be allowed to take the control of the vehicle, even if it is to solve such an important problem as localisation. There are several cases where, for example, the vehicle must continue in carrying out the current task and the active localisation can only be performed after the current task or goal is fully finished or achieved.

6.2.1 Passive Localisation Layer

This module represents the *passive* localisation system of the vehicle, as described in Chapter 3. For the purpose of this thesis, it is based on the improved particle filter algorithm described in Section 3.2. Particle Filters are the chosen technique for the passive layer, as they can be applied in a variety of different scenarios, they don't require an initial state knowledge, and can effectively represent multiple hypotheses, which is definitely the case when similarities arise in the environment and the vehicle can be confused about its true state. Additionally particles can easily be grouped into clusters to reduce the number of possible states, and already defined formulations easily provide information about their distribution and entropy.

6.2.2 Formulation

The active localisation problem can be defined as a planning problem, expressed by the tuple $\Psi = (Bel(x_{ti}); \mathbb{A}; \gamma; \delta)$ where:

- $Bel(x_{ti})$ represents the robot belief of its state at time t_i, i.e. the probability distribution function;

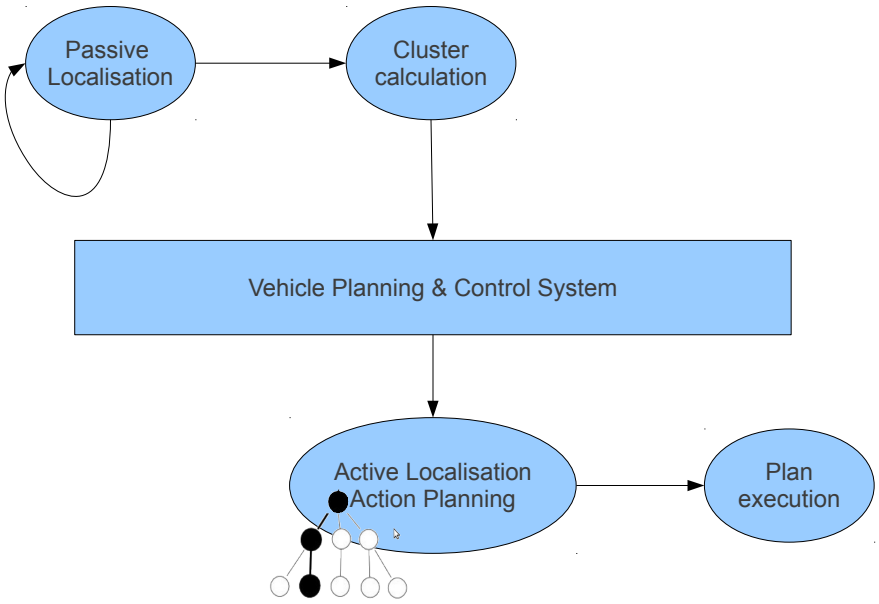


Figure 6.1: The general architecture of the navigation system: the passive localisation module is always running. According to the probability distribution of the state, the particles are clustered and centroids are calculated. According to the entropy, cluster features and plan constraints, the active localisation module can be triggered. Through an exploration of the tree structure, it outputs the set of actions to be executed by the vehicle.

- $\mathbb{A}$ represents a set of n possible actions $\{a_1, a_2, ..., a_n\}$
- γ represents the cost associated to the execution of the actions. In case the cost of each action are independent from previous actions, it is a vector of n elements, associated to the n actions. On the other hand, if the cost is not independent, it is represented by a more complex function.
- δ represents the reward associated to the successful completion of the action. It is not represented by a fixed value or a vector, but it is a function of the belief on the state.

The output of the algorithm is represented by a set of s actions $\mathbb{S} = \{a_{i1}, a_{i2}, ..., a_{is}\}$, where $a_i \in \mathbb{A}$, which maximises the function $\sum_i [c_{\delta,i}\delta_i(Bel(x_{t0})) - c_{\gamma,i}\gamma_i(Bel(x_{t0}))]$, where $c_{\delta,i}$ and $c_{\gamma,i}$ represent weights in order to be able to give more or less importance to specific (order of) actions.

Particle filters by definition represent a discrete representation of the probability distribution function. Each particle represents a possible state of the robot, and therefore a possible hypothesis. However, for computational reasons, there is an interest in grouping the particles which are in the same neighbourhood.

A particle p_i belongs to a cluster C_j, with centroid c_j if the distance is smaller than a threshold $|p_i - c_j| < \vartheta_p$, with $\vartheta_p > 0$

A cluster C_i is considered *compact* if the entropy of the particles belonging to the cluster is smaller than a threshold $H(C_i) < \vartheta_C$, with $\vartheta_C > 0$

Following this formulation, it is possible now to analyse the proposed approach more in details.

6.2.3 Cluster Calculation

Active localisation is useful to discriminate among several hypotheses for the state estimation. Although it could literally start taking as input the total amount of particles, the problem would easily become intractable, due to the required computational power. Additionally, many particles surrounding the same state do not add valuable information, in terms of possible hypotheses. On the other hand, they give an estimate of the probability of the same singular hypothesis. For those reasons, it is useful to proceed with a cluster analysis, to identify a few hypotheses, rather than working with hundreds or even thousands of possible states. The algorithm needs to be generic enough to allow the vehicle to discriminate between n possible states. The number of possible states n is not known in advance, but it is bounded $0 < n \leqslant T$, with $T \in \mathbb{N}$. This means that there is an assumption on the maximum number of

states, for computational reasons. The proposed system therefore iterates the cluster calculation through an iterative *k-means* algorithm. As shown in algorithm 1, at each iteration, the algorithm checks if the clusters are *compact*, according to definition 6.2.2. If all the clusters are *compact*, this information is passed to the Vehicle's Planning and Control System, to enable active localisation, as in Figure 6.1.

Data: $Bel(x_{ti})$ (i.e. particles), T
Result: C (set with clusters), trigger for active localisation
$nCluster = 1$;
while $nCluster \leqslant T$ **do**
    $C = kmeans(Bel(x_{ti}, nCluster)$;
    compact = true;
    foreach *cluster* C_i *in* C **do**
    if *!compact*(C_i) **then**
        compact=false; break;
    end
    if *compact* **then**
        notifyToPlanner();
    end
end

Algorithm 1: Clusterisation of the particles for active localisation. The planner is notified only when the computed clusters are compact.

6.2.4 Vehicle's Planning and Control System

This module receives the information from the passive localisation system, when there is a clear clusterisation, i.e. all the clusters are *compact*, according to definition 6.2.2. The goal is to enable the active localisation only when it is needed and when it does not interfere with the current vehicle's goals, if their accomplishment is more important than performing a different set of action aiming at localisation. For example, if the vehicle is performing a mission-critical inspection of an underwater structure, it might be more advantageous to continue the inspection using relative position with respect to the structure. In other cases, it would be better to clarify its current location, in order to avoid keeping inspecting possibly the wrong structure. A detailed analysis of this module is however out of the scope of this thesis, as its focus is on the localisation mechanisms. In stand-alone tests of active localisation, this system was always triggering the active localisation module, when receiving information about the clusterisation, with more than one cluster. In this way, it ensures that there is a finite and computationally tractable number of hypothesis over which the system needs to discriminate.

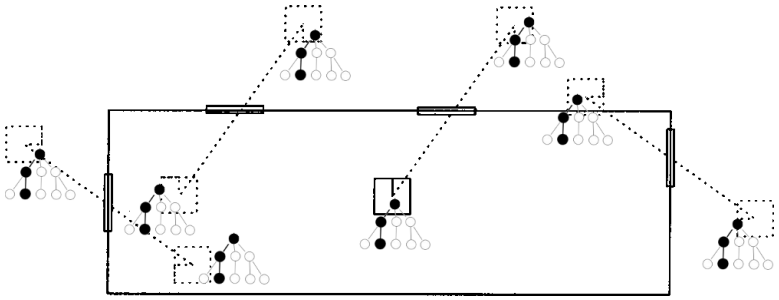


Figure 6.2: From each of the eight possible robot locations, a tree of actions is computed. Analysing the results of the actions across the possible locations, the algorithm will determine the best path in the tree.

6.2.5 Active Localisation Layer

This module is started from the Vehicle's Planning and Control System and has the task of producing a set of action, with the goal of minimising the entropy of the expectation in the particle distribution and contribute therefore to an effective localisation. At this stage, the full particle distribution is not considered any more, but only the centroids of the clusters are considered as possible hypothesis. This module, as described in section 6.2.2 produces a set of actions, whose execution will help discriminate among the different clusters. Basic actions a_i that the vehicle can perform are identified:

$$\mathbb{A} = \{a_1; a_2; ...; a_n\} \tag{6.1}$$

The actions a_i are on the format: "*go forward for x meters*", "*go backwards for x meters*", "*go left for x meters*", "*go right for x meters*", "*go up for x meters*", "*go down for x meters*", "*turn x* deg *clockwise*", "*turn x* deg *anticlockwise*".

The module produces a list $a_{t_0}, ..., a_{t_s}$ which represents the s actions selected to be executed at times $t_0, ..., t_s$.

The proposed approach is to build a tree of basic actions from each cluster centroids, as shown in Figure 6.2. Figure 6.3 shows a tree built on a possible location (cluster i), with the root of the tree initialised at the centroid, and with four basic actions. The complexity of the tree exploration is polynomial on the number n of actions and exponential on the depth d of the tree ($O(n^d)$). However, it is possible to reduce this complexity, considering that for every basic action a^i there is another basic action a^j which produces the opposite effect. As visiting a location already visited is not providing any new information, each node will not expand the node with the action which balances the previous one, as shown in Figure 6.4. Following the same principle, loops on the same root-to-node path are not allowed, thus reducing the final complexity, as shown in Figure 6.5. This means to avoid visiting the same state more than once, as also suggested by Jensfelt & Kristensen [47]. Table 6.1 shows the

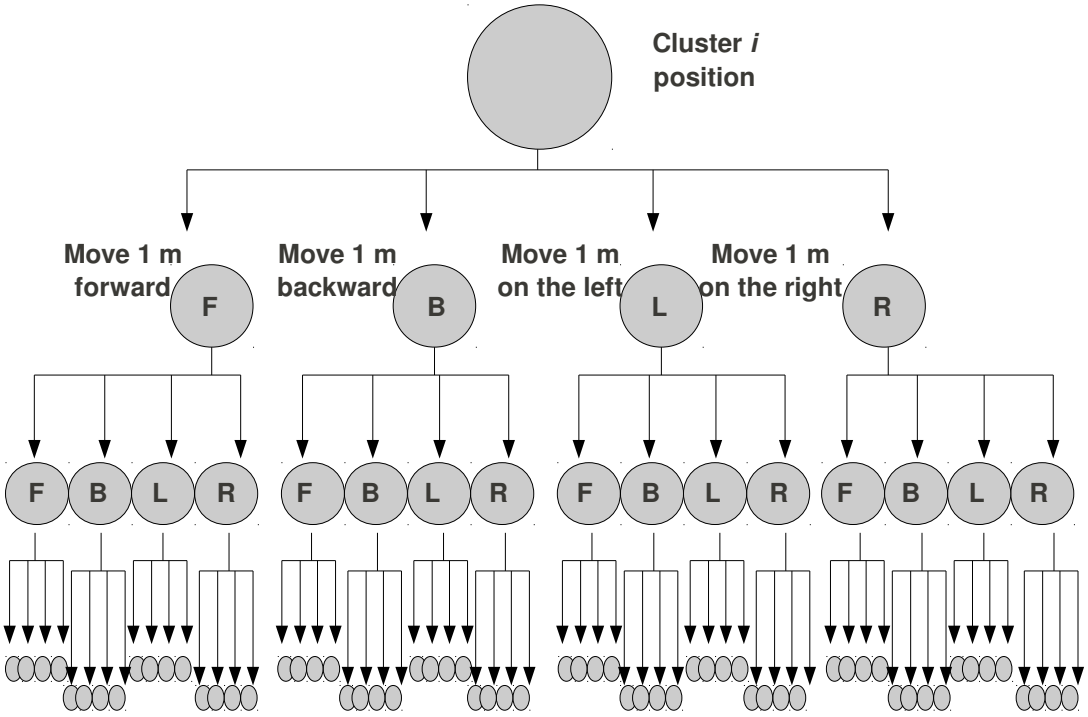


Figure 6.3: An example of a tree built from the centroid of cluster i, with four basic actions/movement.

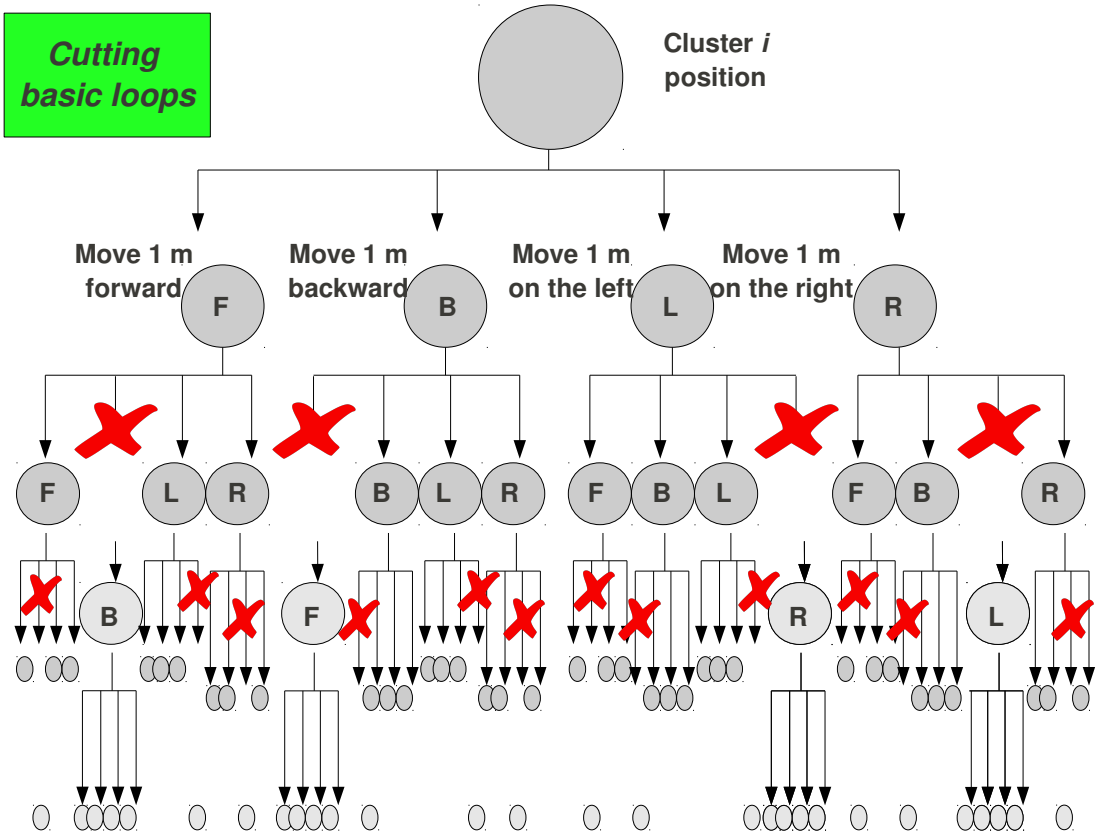


Figure 6.4: First optimisation step: cutting basic loops.

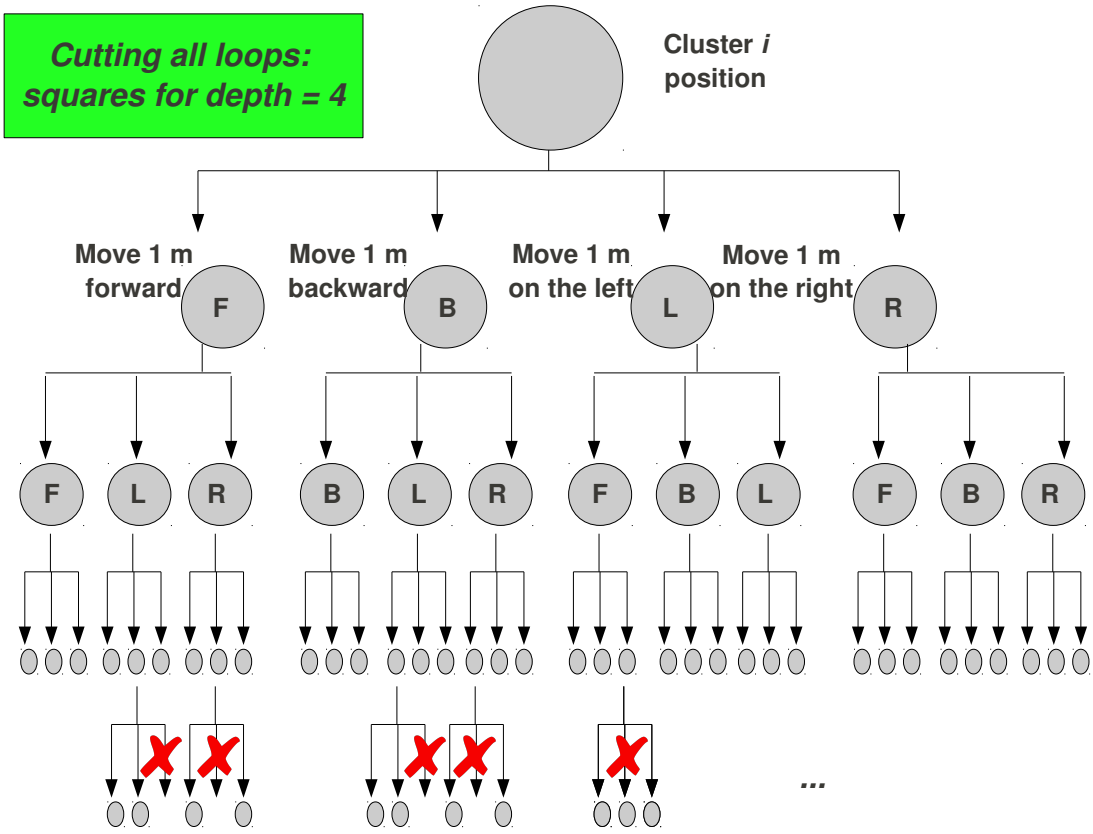


Figure 6.5: Second optimisation step: cutting all loops.

effects of the optimisations considering four basic actions, highlighting the advantages in terms of nodes to be visited according to the depth and the chosen algorithm.

It is also possible to set other constraints, in order to cut the tree. They can be related to the specific vehicle used, and to some manoeuvres which should be avoided. For simplicity, no other constraints are considered, but they are easily pluggable in the module. Modelling the robot behaviour as a set of basic actions is very important in order to consider that different paths to the same location can produce a different probability density function of the vehicle state. Another possibility would have been to consider longer trajectories, instead of basic actions. First of all, choosing to model trajectories would have limited the framework to trajectories only, whilst it has been designed to account to any type of actions, not only movements. Additionally, in order to select among several trajectories, an action/cost associated to the trajectories would have been needed and therefore they would have been discretised to analyse several points of the trajectories. Finally, several different types of trajectories could be represented, like for example spiral, straight-line, lawn-mower, etc. and it would have been difficult to create a stardardised approach for the discretisation. In complex environments a very specific trajectory - not directly linked to a trajectory family,

depth	# actions no optimisation	# actions cutting basic loops	# actions cutting all loops
1	4	4	4
2	20	16	16
3	84	52	52
4	340	160	152
5	1,364	484	436
6	5,460	1,456	1,216
7	21,844	4,372	3,388
8	87,380	13,120	9,304
9	349,524	39,364	25,572
10	1,398,100	118,096	69,672
11	5,592,404	354,292	189,964
12	22,369,620	1,062,880	514,896
complexity d = depth	4^d	3^d	2.724542468^d

Table 6.1: Number of nodes and complexity in function of the depth of the tree, showing the benefits of the optimisations in the tree exploration.

could be the one needed for localisation. On the other hand, basic moves are like bricks which allow to build any of those trajectories, in a discretised way. Basic actions however needs to ensure the robot safety. In the next section a brief explanation of the obstacle avoidance system will be presented.

Dealing with obstacles

A key safety constraint for the robot is to avoid obstacles whilst performing a task. This is very relevant for active localisation as well, because the control of the robot is in the navigation loop and specific sets of actions are chosen to solve the localisation problem. Considering that the map and the possible robot locations are known in advance, the only trajectories that can be generated are those who satisfy safety constraints for all the possible locations. This has the effect to further cut branches of the tree, thus making the algorithm even more efficient. Trajectories which would bring the robot close to obstacles for one cluster and in free space for another cluster are to be preferred, as it will be explained more in details in the next section, about action rewards. However, this is done directly in the node evaluation, without explicitly considering obstacles in the reward calculation. In the next sections a proposal of action reward and cost will be presented.

Action Reward

As previously said, it is important to minimise the expectation of the entropy in the probability distribution function. However, the reward function is calculated for each node of the tree, and therefore it can be easily classified as the critical operation for the system. Fairfield & Wettergreen pointed out that this calculation is very time consuming ([30]). The proposed approach therefore tries to avoid it, analysing the meaning of the minimisation of the expected entropy. The proposed approach considers the information gain acquired after executing the actions $a_{t_0}, ..., a_{t_s}$. It is represented by the diversity in the expectation of the future measurements for the different hypothesis. The following notation is used:

- k^{tot} represents the total number of clusters
- k represents the Cluster k;
- n represents the node n in the tree;
- z_n^k represents the measure from node n in cluster k. It is represented as an array of distances;
- m represents the number of cells in the array of measures (i.e. the number of beams in a sonar)
- $z_n^{j,k}$ represents the scalar value of the measure z_n^k at index j, with $1 \leqslant j \leqslant m$

The algorithm seeks nodes to maximise the differences in the measurements from different clusters. Considering that an indication of the difference in a set of numbers is given by the variance σ^2, the reward of a node n can be expressed as:

$$r(n) = \frac{\sum_{j=1}^{m} \sigma^2_{z_n^{j,k=1:k^{tot}}}}{m} \tag{6.2}$$

It represents the average of the variance for each index j of the measurements z_n, acquired from the different clusters k, with k ranging from 1 to k^{tot}. The gain calculation is also explained in Figure 6.6. If we consider only translations and rotations as set of basic actions, each action applies a transformation matrix to the centroid. Considering x the initial state of the centroid, the resulting position x' at node n is therefore given by:

$$x' = [RT]_{a_{t_h}} \cdot [RT]_{a_{t_{h-1}}} \cdot \quad \cdots \quad \cdot [RT]_{a_{t_1}} \cdot x \tag{6.3}$$

where h represents the depth of node n and a_t the action executed at time t.

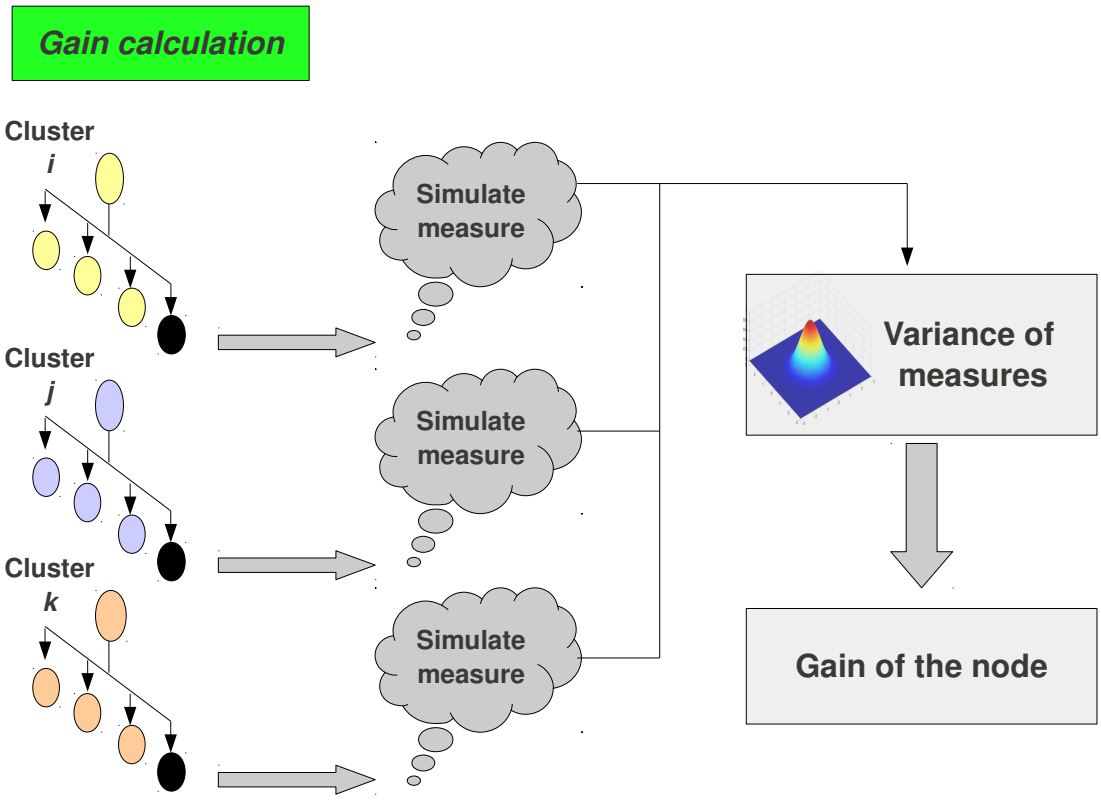


Figure 6.6: The reward calculation for each node of the tree.

Cost of actions

Each action has not just a reward, but also a cost associated to it, which may vary according to the specific constraints. For example, in many cases, it is reasonable to assume that if the action a_t is the same than the action a_{t-1}, the cost for the vehicle would be smaller than for a completely new action. This surely happens when the actions represent vehicle basic moves, because there is no need to radically change the thruster behaviour. The total value therefore assigned to a single node is given by the difference between the reward r and the cost c. The output of the module is therefore:

$$p^* = \{a^*_{t_0}, ..., a^*_{t_s}\} = argmax_{p_i}(r_{p_i} - \alpha * c_{p_i}) \tag{6.4}$$

where p_i represents the path i^{th}, α is a constant representing the scale of the cost with respect to the reward and p^* represents the best path. The design of the cost function has of course a significant impact on the path generated and it can vary according to the robotic platform and to the environment. In the proposed framework, the cost of every action c_{a_i} is calculated as follow:

$$c_{a_i} = hC_i * (1 + h * nI + k * nTZ) \tag{6.5}$$

where hC_i is the hydrodynamic coefficient linked to the specific action a_i, h is the coefficient weighting the nI thrusters requiring an inversion of voltage, in order to execute the action, and k is the coefficient weighting the nTZ thrusters requiring a change of voltage to zero, with $h > k$. For a torpedo shaped slide-capable vehicle, a possible set of parameters is:

- hC_i is equal to 0.6, if a_i represents a forward movement; to 1, if a_i represents a rotation; to 2, if a_i represents a sideways movement);
- h is equal to 1;
- k is equal to 0.5;

Stopping criteria

The stopping criteria for the exploration of the tree are as follow:

1. a predefined depth is reached. In this case the best path is given to the path execution system;
2. the reward for a node has reached a specific value. That means that the path up to that node is able to disambiguate among the different clusters.

The ability to stop the tree exploration at any depth is very important for in-mission executions, where time-constraints do not allow prolonged reasoning.

Can the algorithm guarantee convergence after stopping criteria is reached? The easy answer is not in all cases, like all solutions for robot navigation. Robot navigation is very dependent on the environment and there can be very tricky environments where the disambiguation among several hypotheses is not only difficult, but mathematically impossible. A very easy example to show is two identical environments, A and B, which are not connected. In this case the robot does not have any possibility to understand if he is in the environment A or in the environment B. Another example is a standard rectangular pool, with no compass. Any trajectory or set of actions to sense the environment will not be able to discriminate among the two symmetric solutions. Both examples are related to closed environments. In those cases, even without setting a maximum length of the tree, the algorithm would terminate when it has fully explored the environment and there are no more nodes in the tree to expand. It is possible to easily detect those cases and therefore to notify to the upper vehicle's layer about the impossibility to solve the localisation problem. This happens when the reward for each explored node is below a certain noise threshold, thus detecting practically symmetries in the environments which are impossible to solve. Those cases are however theoretical cases, as in real environments chances of impossible localisation are rare, if not impossible. Those cases are well known in advance, as for localisation problems the map is known in advance. There would not be any robotic mission required accurate localisation if the environment would be known to be an impossible one for localisation.

Summary of the module

Summarising, the general principle of the module is to find a path (or, more in general, a set of actions) that maximises the diversity in the observations from the different initial possible positions. From the centre of each cluster, an action tree is built. Each node represents a possible basic action. The output of the module is a path root-leaf (i.e. a sequence of basic actions) which maximises the diversity in the observations and thus minimising the expected entropy.

6.2.6 Plan Execution

After the best set of actions is chosen, they need to be executed. The results of this execution will minimise the entropy and, through the filtering process, will eventually drop at least one cluster from the hypothesis space, if the problem is solvable. There are cases when the problem is not solvable, i.e. given any set of action, it would still

be impossible for the vehicle to determine its own location. However, those scenarios are generally far from the real applications. They would be nevertheless detected by the tree-based planner, as no path/node in the tree would give a better gain/cost compared to others.

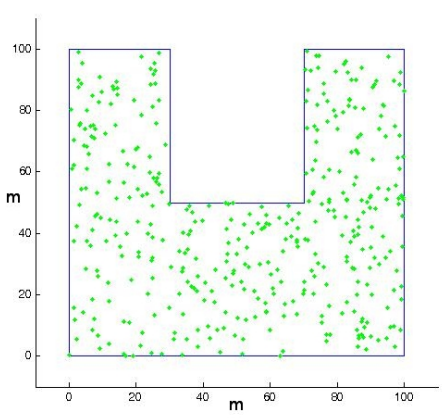
6.3 Experimental Results

6.3.1 Simulated setup

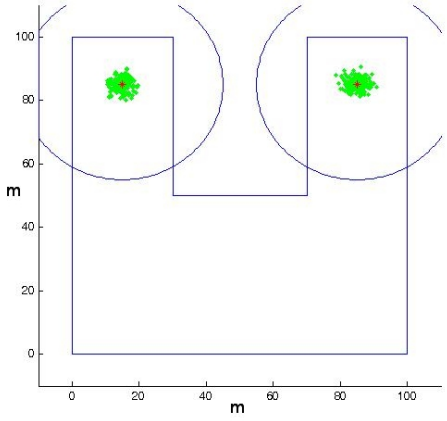
The algorithm has been tested in a simulated setup first. For simplicity, only 2D environments have been tested, in order to reduce the set of basic moves. The 3D extension is very straightforward, as the only change is the number of elements in the set of possible actions. Conceptually and practically, there is no difference, apart from the computational time, which is however relatively low. The sensor modelled is a Tritech Micron, currently mounted on our vehicle *Nessie IV*. It is a mechanically scanning imaging sonar (MSIS), with 360 deg field of view. For this reason, in this first setup there are no rotations in our set of basic moves, as they do not provide more information about the environment. In the case in which the field of view is limited, then the rotation basic actions are necessary. The set of basic actions is thus represented by:

$$\begin{aligned} \mathbb{A} = \text{move}\{&\text{forward, backwards, left, right}\} \\ &\text{for 6 meters} \end{aligned} \tag{6.6}$$

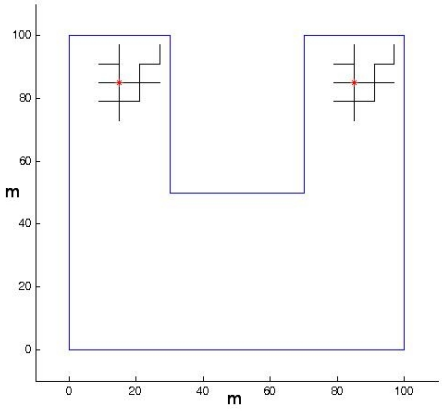
The first environment has a *U configuration*, described in Figure 6.7. The vehicle is either at the end of one of the two legs of the *U*. With the vehicle positioned on the left leg, the particles, initially spread all over the environment, quickly converge to two possible symmetric locations. Of course, the same result appears with the vehicle positioned on the right leg. As the observation from the two points is the same, it is not possible to distinguish between the two hypotheses with classical passive techniques. The control is then given to the active localisation module which takes the two locations of the cluster centroids in input. The dimensions of the environment are 100x100 meters, with each leg long 50 meters and 30 meters wide. The two cluster centroids are located at $(15; 85)$ and at $(85; 85)$ With a sonar range of 40 meters, the output of the module is a path composed by six basic moves, all going backwards. This is actually the best path in order to discriminate between the two solutions, as the diversity in the environment can be sensed on the bottom of the *U*. Reducing the range to 30 meters, the output is composed by eight basic moves: six backwards and then two on the left. This is consistent with the expectations, as the generated path



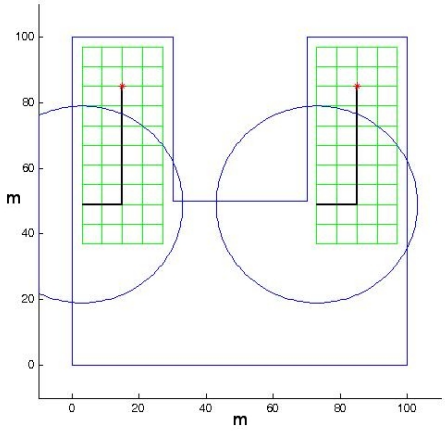
(a) Initial state: uniform distribution of the vehicle state



(b) Clusterisation of the particle



(c) Tree built from each centroid



(d) Path chosen to maximise the information gain

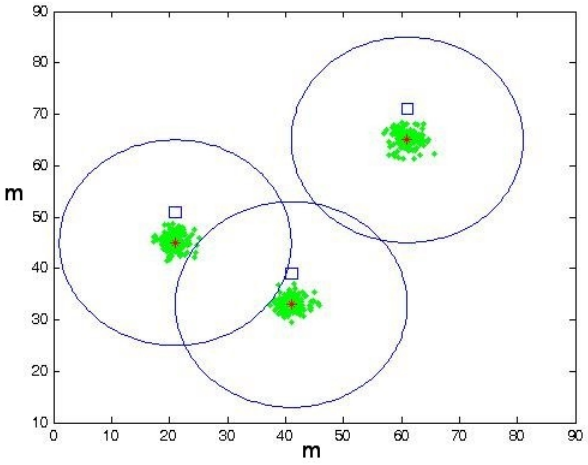
Figure 6.7: All the steps of the active localisation process: clusterisation, tree construction and path building. First scenario: U-like closed environment

Environment	Basic Actions	N. Clusters	Sonar Range	Sonar FOV	Path generated
U-shaped 100x100m	$\{F, B, L, R\} for$ 6m	2	40 m	360deg	B - B - B - B - B - B
U-shaped 100x100m	$\{F, B, L, R\} for$ 6m	2	30 m	360deg	B - B - B - B - B - B - L - L

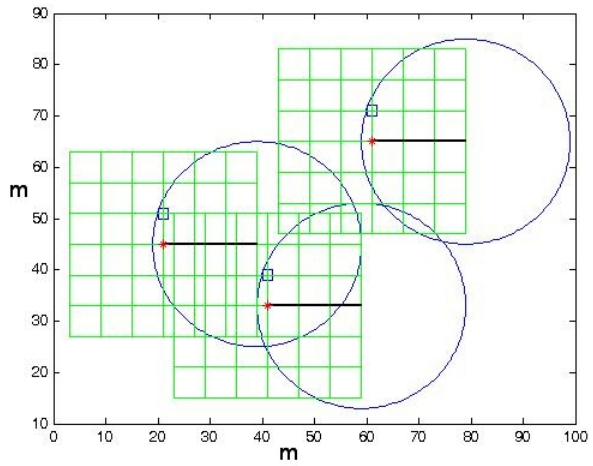
Table 6.2: Active localisation in U-environment. Two clusters at (15;85) and (85,85). According to the sonar range, the generated path is different. If less information are available (reduced range) the path is longer.

arrived to a point very close to the borders of the environment (presence of obstacles) for one cluster, while for the other cluster, the final point was far from any obstacle. This simulated environment shows the possibility to apply active localisation in closed environment, like it is often the case for man-made environments, like marinas. All the steps are highlighted in Figure 6.7. Table 6.2 summarises the results.

The second environment is more similar to open sea conditions and leading to offshore applications. There are no boundaries, just three objects, which can represent an underwater site, as in Figure 6.8. Assuming that the vehicle is travelling from one site to another one, it is very likely that the navigation error is bigger than the distance between two objects and thus the vehicle needs to find a way to discriminate between the initial hypothesis (in this case, three). This case is also interesting as it shows how a small change in the parameters of the sonar can change significantly the results. Due to the location of the underwater objects, a range of the sonar over 27 meters can discriminate between the positions without need of any active localisation. Reducing the range gradually, the generated path change significantly. Between 26 and 27 meters, one move is enough to distinguish between the three hypotheses and the selected move is to go backwards. For the top right centroid, this has the effect to go nearer the two middle objects. When the range drops up to 23 meters, the selected move is to go right. For the left centroid, this has the effect to go nearer the central object. Reducing again the range, the required trajectory is composed by two steps and again the first choice (up to 21 meters) is to go backwards, creating a measure discrepancy between the top centroid and the other two. At 20 metres range, the two steps are on the right. Up to 17 meters range, the planned path is to go on the right for three steps: this helps to discriminate the left centroid (very near to the central object) with respect to the other two objects. It is now interesting to see what happens for sonar range below 17 meters. There is no straight exit from the Active Localisation module, so the tree is fully explored until the maximum depth (fixed at nine). However, the best discriminant path that the algorithm can found is not a



(a) Particle clustered in front of the three objects



(b) Path chosen to maximise the information gain

Figure 6.8: Second scenario: three objects in an open environment.

Environment	**Basic Actions**	**N. Clusters**	**Sonar Range**	**Sonar FOV**	**Path generated**
3 objects, 100x100m	$\{F, B, L, R\} for 6$m	3	$> 27\ m$	360deg	-
3 objects, 100x100m	$\{F, B, L, R\} for 6$m	3	$26 - 27\ m$	360deg	B
3 objects, 100x100m	$\{F, B, L, R\} for 6$m	3	$23 - 25\ m$	360deg	R
3 objects, 100x100m	$\{F, B, L, R\} for 6$m	3	$21 - 22\ m$	360deg	B - B
3 objects, 100x100m	$\{F, B, L, R\} for 6$m	3	$20\ m$	360deg	R - R
3 objects, 100x100m	$\{F, B, L, R\} for 6$m	3	$15 - 19\ m$	360deg	R - R - R

Table 6.3: Active localisation in an open environment with three objects. The variation of the sonar range has the effect to variate the generated path.

path of length 9, but it is the last path generated for a sonar range of 17-20 meters, of length 3. Table 6.3 summarises the results. A representation of the environment, with particle clustering and chosen path is highlighted in Figure 6.8.

A complex simulated test - a labyrinth-style environment, Figure 6.9 - represents our third simulated environment. For such complex scenario with so many constraints, given by the walls, the algorithm needs to be applied iteratively, in order to arrive to a final unique determination of the robot pose. The first path generated by the algorithm drops the number of clusters from the initial six to three. The second path drops it from three to two, while the third path provides a unique solution. It is to be noted that the last path is actually a degenerated path, with the robot deciding not to move. This decision is very rare, if not impossible, at the very beginning of the active localisation module, otherwise it would mean that two (or more) different positions sensing substantial different measures would have all a certain likelihood to represent the robot pose. It is however possible when there is an iteration of the algorithm, like in this case. The second path has dropped one cluster (the fourth), while keeping the position is recognised to be the best action to discriminate between the last two clusters. Table 6.4 summarises the results.

Finally, a fourth simulated test has been performed, changing the parameters of the algorithm in order to model actions and cost related to both a different vehicle with different sensor capabilities. This was very important to test the portability of the proposed system, which is not bounded to a specific narrow solution, but can represent an important, sometimes vital, tool in any system. The vehicle actions were therefore modelled considering the vehicle *Nessie V*. This vehicle is fully actuated in five degrees

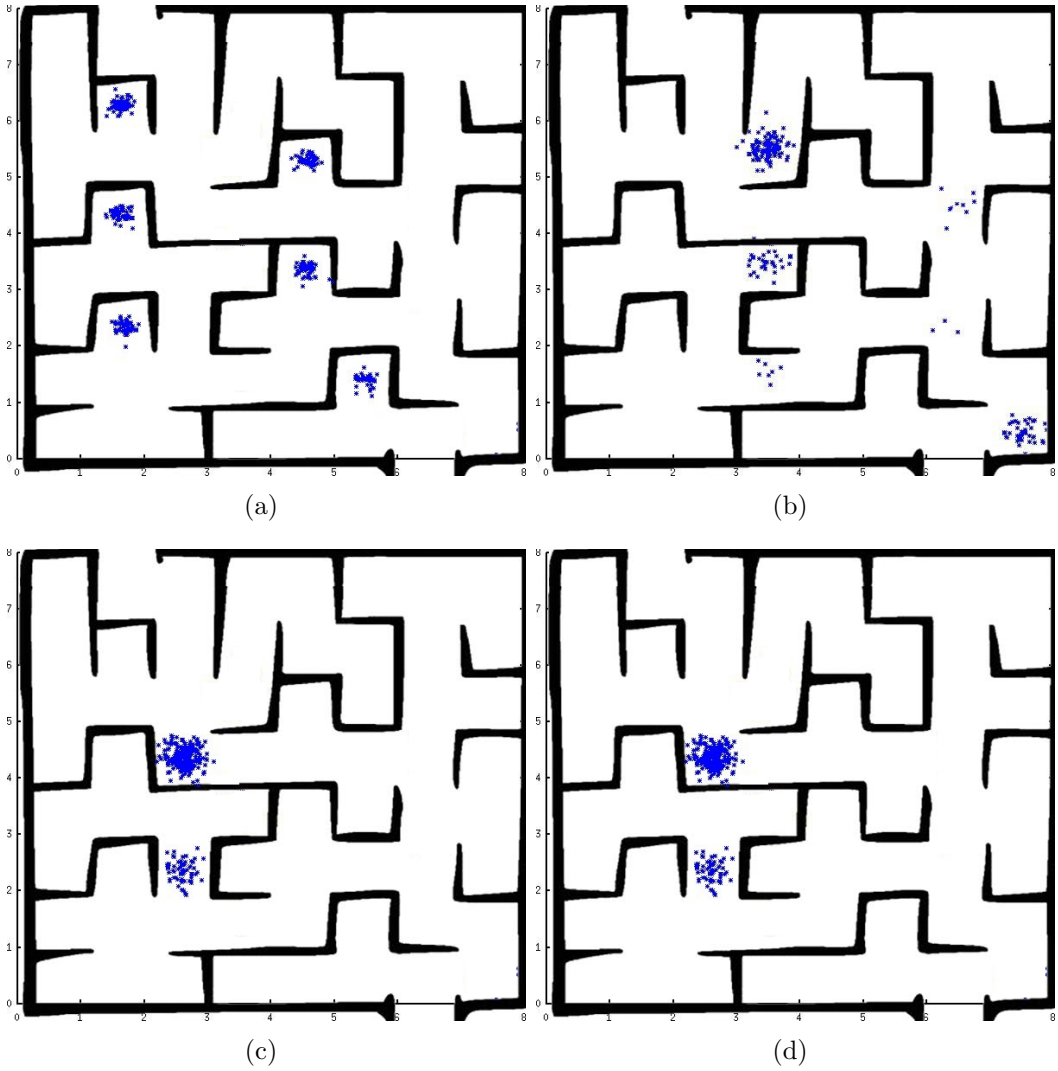


Figure 6.9: Labyrinth environment: 80x80 m, with each square of 10 m. Sonar range set to 10 m. (a) initial distribution, with six possible locations; (b) pose estimation after first path (down-right-right) is executed. Three clusters are dropped; (c) pose estimation after second path is executed (left-down). An additional cluster is dropped; (d) execution of the third path (stay still): particle convergence.

Iteration	Basic Actions	N. Clusters	Sonar Range	Sonar FOV	Path generated
1	$\{F, B, L, R\} for$ 10m	6	10 m	360deg	B - R - R
2	$\{F, B, L, R\} for$ 10m	3	10 m	360deg	L - B
3	$\{F, B, L, R\} for$ 10m	2	10 m	360deg	-

Table 6.4: Active localisation in a labyrinth environment. Initially six clusters are identified. The first path is generated (B-R-R). The filter drops three clusters. The active localisation is again performed and a new path generated (L-B). After this, the particles can be grouped in two clusters only. At this stage a new path is computed, but the algorithm does not provide any new actions. Remaining in that location, the filtering processing drops one of the two clusters and shows the real location of the robot.

of freedom (*surge*, *sway*, *heave*, *pitch*, *yaw*), so one more (*pitch*) than *Nessie IV*, a different set of actions was therefore chosen, linked to the sensor mounted. *Nessie V* is equipped with a multibeam Tritech Gemini sonar. Although this sensor is very precise (256 beams), and very fast in delivering images, it has a limited field of view of 120 deg. Due to this constraint, it is necessary to incorporate basic rotation actions in the framework. On the other hand, some other basic actions, like some considered in the examples previously explained, cannot be safely executed, due to the limited field of view. A conservative safe behaviour was therefore chosen, resulting in this set of basic actions:

$$\begin{aligned} \mathbb{A} = \{&\text{move forward for } w \text{ meters;} \\ &\text{rotate } q \text{ degrees}\{\text{clockwise, anticlockwise}\}\} \end{aligned} \tag{6.7}$$

This configuration was tested on the the first simulation scenario, based on the U-environment, already defined in Figure 6.7. w was set to 6m, as in the previous simulation. Two tests were performed with q equal to 90 deg first and to 60 deg. The path generated in output was a double rotation of 90 deg clockwise and then five steps of going forward. In the second case, the generated path was very similar, i.e. three rotations of 60 deg and then five steps forward. Also in this case, it is possible to see that the vehicle's choice is to target an area of diversity, starting from the two clusters, which is represented by the start of the U-legs in this scenario. Comparing the path generated in output with the path previously generated for the *Nessie IV* vehicle, both similarities and differences can be identified. Both algorithms made the vehicle navigating the U-leg towards the main area. *Nessie IV* also decided to make two steps lateral, in order to drop one of the cluster quicker, whilst *Nessie V* decided that it was enough just to rotate and navigate towards the beginning of the

Environment	Basic Actions	N. Clusters	Sonar Range	Sonar FOV	Path generated
U-shaped 100x100m	F for $6m$; $\{CR, ACR\}90$ deg	2	40 m	120deg	CR - CR - F - F - F - F - F
U-shaped 100x100m	F for $6m$; $\{CR, ACR\}60$ deg	2	40 m	120deg	CR - CR - CR - F - F - F - F - F

Table 6.5: Active localisation in U-environment. Two clusters at (15;85) and (85,85). The vehicle chooses to rotate and navigate towards the main tank area.

leg. This apparent behaviour difference is linked to the field of view of the sonar. In the first case, the lateral movement was useful to radically differentiate among the two clusters, whilst with a limited field of view, this was considered unnecessary. It is also interesting to analyse the fact that, with different rotation actions, the same path was given by the algorithm. This is due to the gain/cost function. Surely a rotation of 60 deg before reaching the end of the leg could be enough to differentiate among the two clusters. However, the cost of the forward movement is significantly smaller than the cost of rotation. Again, it is useful to underline that the clockwise and anticlockwise rotations have the same ratio cost/benefit. The vehicle did choose to rotate clockwise, but simply because it was the first node to be expanded in the tree. Similarly, for the first presented test, the vehicle did choose to move on the left, at the end of the path. A movement to the right would have produced the same exact results. Table 6.5 summarises the results of the fourth simulation scenario.

6.3.2 Comparison with other techniques

The proposed technique performs more reliably than other approaches to the active localisation problem. In this section we analyse the first scenario presented in the previous section, with a *U configuration*, described in Figure 6.7. The vehicle is either at the end of one of the two legs of the U. The set of basic actions is the same than previously described:

$$\mathbb{A} = \text{move}\{\text{forward, backwards, left, right}\} \text{ for 6 meters} \tag{6.8}$$

Random move

The first set of experiments are performed comparing the proposed approach with a random move approach. Each action has the same probability to be selected. Table 6.6 summarises the results of this test, with a sonar of $30m$ and $40m$. The random test

Env.	Basic Actions	Sonar Range FOV	RM average	RM best	AL	AL/RM average
U-shaped 100x100m	$\{F, B, L, R\}$ for 6m	$40m$ 360deg	85.35 steps	6 steps	6 steps	14.22
U-shaped 100x100m	$\{F, B, L, R\}for$ 6m	$30m$ 360deg	113.18 steps	8 steps	8 steps	14.15

Table 6.6: Comparison with composition of random moves. The vehicle is able to localise itself more than 14 times faster than the average of a random move trajectory.

Env.	Basic Actions	Sonar Range FOV	RM average	RM best	AL	AL/RM average
U-shaped 100x100m	$\{F, B, L, R\}$ for 6m	$40m$ 360deg	41.82 steps	6 steps	6 steps	6.97
U-shaped 100x100m	$\{F, B, L, R\}for$ 6m	$30m$ 360deg	54.52 steps	8 steps	8 steps	6.82

Table 6.7: Comparison with composition of random moves avoiding to choose an action with opposite effect of the previous action. The vehicle is able to localise itself almost 7 times faster than the average of a random move trajectory.

has been performed 10,000 times for each sonar parameter and in the table both the average length of the path and the best path found in 10,000 runs are shown. The best path found in 10,000 runs has the same length than the one computed by the active localisation module, whilst the average path is more than 14 times longer in both cases. The worst path constituted of 1,502 and 1,043 single steps, respectively, being it 174 and 187 times worst than the proposed approach.

Random move avoiding basic loops

The second analysis presented uses the same approach of selecting a random move, but avoiding to choose an action which would be the opposite of the previous action. For example, if the current action selected randomly makes the robot moving forward, then the following action cannot be a backward movement. Table 6.7 summarises the results of this test, with a sonar of $30m$ and $40m$. The random test, with selective random action, has been performed 10,000 times for each sonar parameter and in the table both the average length of the path and the best path found in 10,000 runs are shown. The best path found in 10,000 runs has the same length than the one computed by the active localisation module, whilst the average path is almost 7 times longer in both cases. The worst path constituted of 642 and 534 single steps, respectively, being it 80 and 89 times worst than the proposed approach.

Env.	Basic Actions	Sonar Range FOV	RM average	RM best	AL	AL/RM average
U-shaped 100x100m	$\{F, B, L, R\}$ for 6m	$40m$ 360deg	15.06 steps	6 steps	6 steps	2.51
U-shaped 100x100m	$\{F, B, L, R\}for$ 6m	$30m$ 360deg	17.50 steps	8 steps	8 steps	2.19

Table 6.8: Comparison with composition of random moves avoiding to choose an action which would bring the robot in an already visited location. The vehicle is able to localise itself more than 2 times faster than the average of a random move trajectory. It is to be noticed that this statistic are only related to the generated paths who allowed the robot to localise itself, as in more than half of the runs, the robot was not able to localise itself following this specific random move trajectory.

Random move avoiding all loops

This approach is based on the selection of a random move which will bring the robot into an explored area. The reason behind this optimisation is the same than for the previous section: an already visited location would not add any new information for the filter. However this approach cannot be used reliably. As the robot executes each action after selection, it can easily arrive to point where no available action can be selected. This is the case for the U scenario analysed. The impossibility to go into already visited locations would prevent the robot to correct a completely wrong path. Although this approach cannot be used with the real vehicle, it has been tested in simulation, again with 10,000 runs for each sonar parameter. Table 6.8 summarises the results of this test, with a sonar of $30m$ and $40m$. In both cases, more than half of the runs (5,188 and 5,953) ended without finding a solution to the localisation problem, arriving to a point where the robot was surrounded by visited locations. Analysing the remaining runs which allowed the robot to correctly localise itself, the best path has the same length than the one computed by the active localisation module, whilst the average path is more than 2 times longer in both cases. The worst path constituted of 41 and 35 single steps, respectively, being it 5 and 6 times worst than the proposed approach.

Selection of the one best action

This case analyses the selection of one best action, and possibly a trajectory given by the sum of n best actions, with respect to our proposed approach. Considering the same U scenario, the selection of one action cannot even be applied. If the action represents a movement up to $15m$, it will not be possible to discriminate among the actions, as for both clusters the robot would sense the same exact information (plus

Figure 6.10: The autonomous underwater vehicle Nessie IV in the OSL tank. It is to be noted the division panel, in order to create two identical environment sections.

noise). This would mean essentially to use a random action approach, discussed in the previous sections. On the other hand, if the basic action is represented by a movement longer than $15m$, the only available action is represented by the action of going backward, as all other actions would have the result to bring the robot out of the map, thus violating obstacle avoidance checks. After the execution of the only available action, however, the robot would still not be able to localise itself, because it has not arrived to a point discriminative enough. In that case, the concatenation of two consecutive actions - the only available - of moving backward of $15m$ would allow the robot to localise itself. This however cannot be generalised. The length of a single step needs to be adjusted and it is not something the robot can easily do. As discussed in the previous Chapter, there are also cases and examples where the concatenation of one best action is not enough for the robot to localise itself.

6.3.3 Tank Trials

The algorithm has been successfully tested in several tank trials, using the facilities at Heriot-Watt University. The test platform was the Autonomous Underwater Vehicle *Nessie IV* [71], equipped with a Tritech Micron sonar to sense the environment, and Doppler Velocity Log (DVL) for motion estimation. A first set of tests has taken place in a 3x4 m tank, as in Figure 6.10.

A panel has been put into the tank in order to create two identical parts in the environment, similar to the *U-scenario* described in the simulated setup. The results are highlighted in Figure 6.11.

The second set of tests has taken place in a 10x12 m tank. The environment is composed by the tank walls plus four panels making two identical sections. The

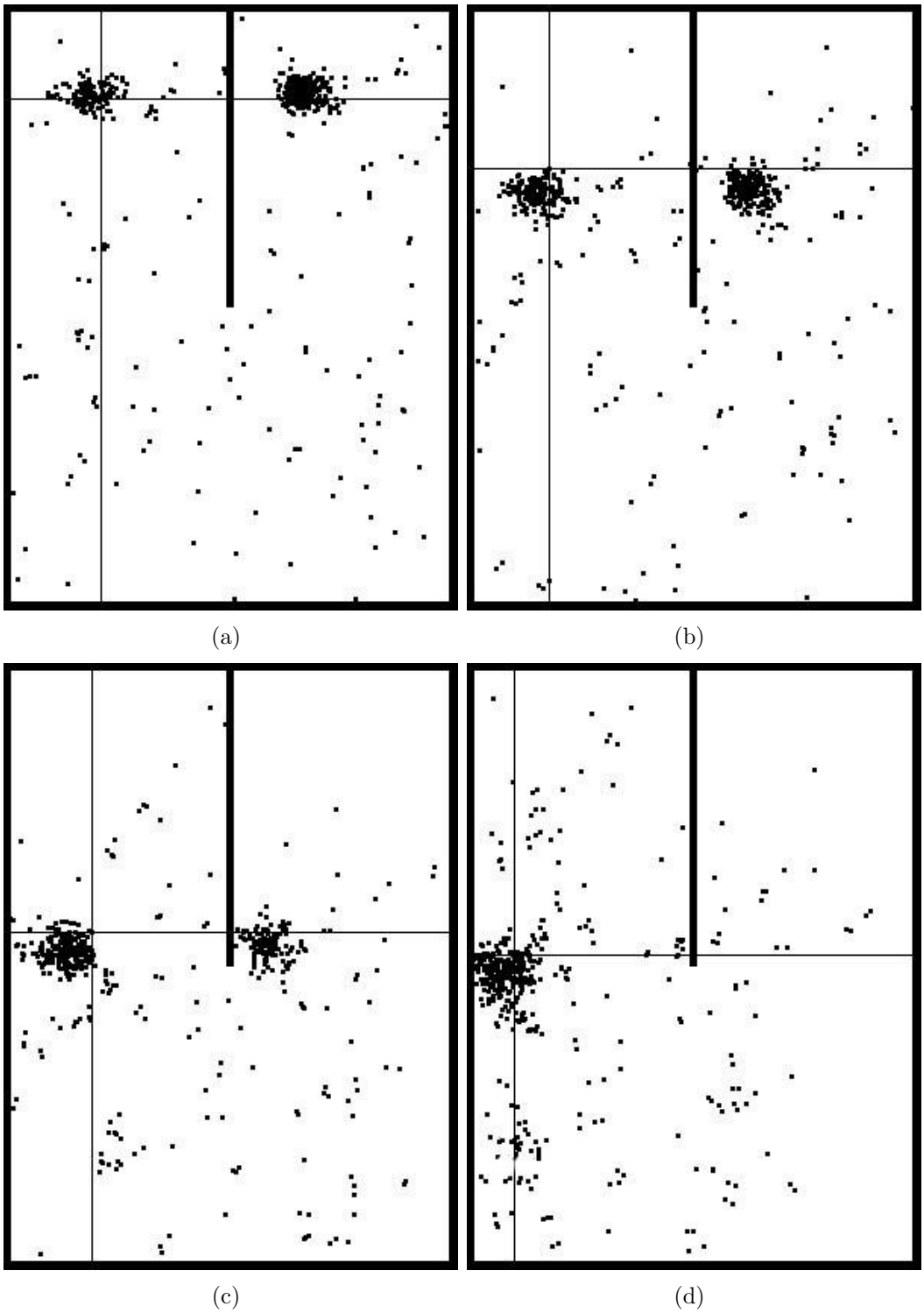


Figure 6.11: Active localisation in the OSL tank, 3x4 m. (a) Initial distribution (b-c) executing the path (down-down-down-down-left-left) (d) Convergence after execution of the path generated by the Active Localisation module.

robot starts with no initial knowledge of its position, so particles are spread over all the environment. After a clear clustering of the particles is reached and stable over time, the active localisation module is triggered and the vehicle computes the path to be executed. After executing the path, the vehicle's position is determined without doubts. All the steps are highlighted in Figure 6.12 and in Figure 6.13.

6.4 Conclusions

This Chapter has presented a novel approach to AUV active localisation. The system has been designed to:

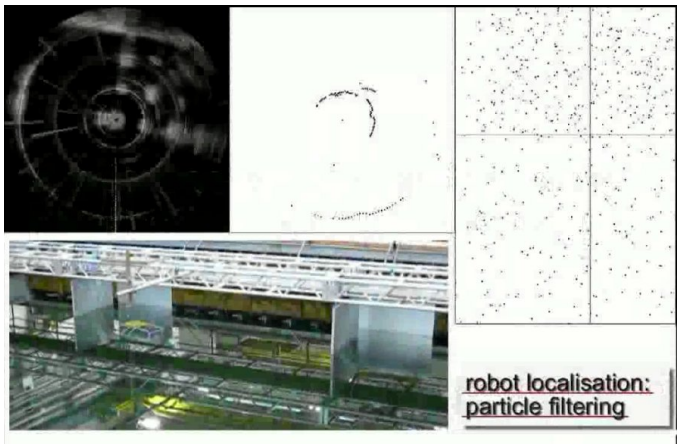
- be general enough, and not tight to any specific custom problem;
- be able to handle multiple possible locations (global localisation), and not just the current one (position tracking);
- consider the entropy expectation as an important information in the definition of the actions to be performed;
- consider multiple actions, as one action is not enough to properly address the problem;
- optimise the computational load

Based on particle filters, the *active* module is triggered when there is a clear clustering of the particle, in order to disambiguate among several hypotheses. From each centroid, a tree of actions is built, and each node is evaluated. The algorithm tries and finds the best node for which measures from the different clusters are most different. This represents the maximisation of the information gain, expressed as the variance of the measurements. Essentially, the algorithm tries to go in places where the sensor measures can help disambiguate among the hypotheses.

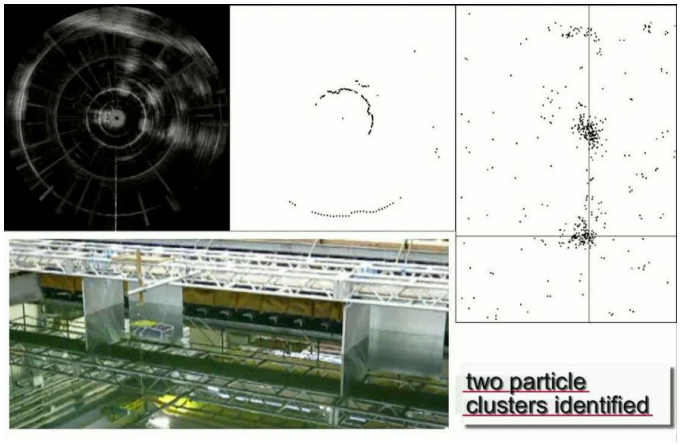
Experimental results both in simulation and in two in-water facilities were presented. It is important to see the different paths generated in the same scenario, changing some of the configuration parameters, such as the sonar range or the basic actions.

In all tests performed, the vehicle is able to successfully localise itself after the execution of the set of actions produced by the proposed algorithm. In one scenario it needed to apply the algorithm iteratively, scaling down the number of clusters at each iteration.

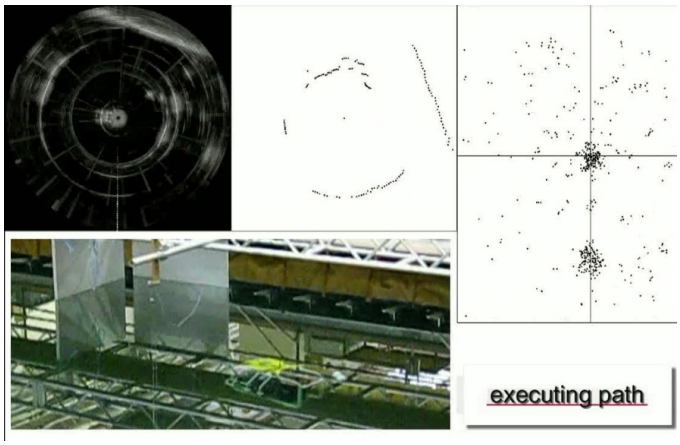
As described in the Chapter, the complexity of the proposed algorithm is related to the number of basic actions, with the tree growing with the growth of the set of



(a) Initial situation, the robot doesn't know its position

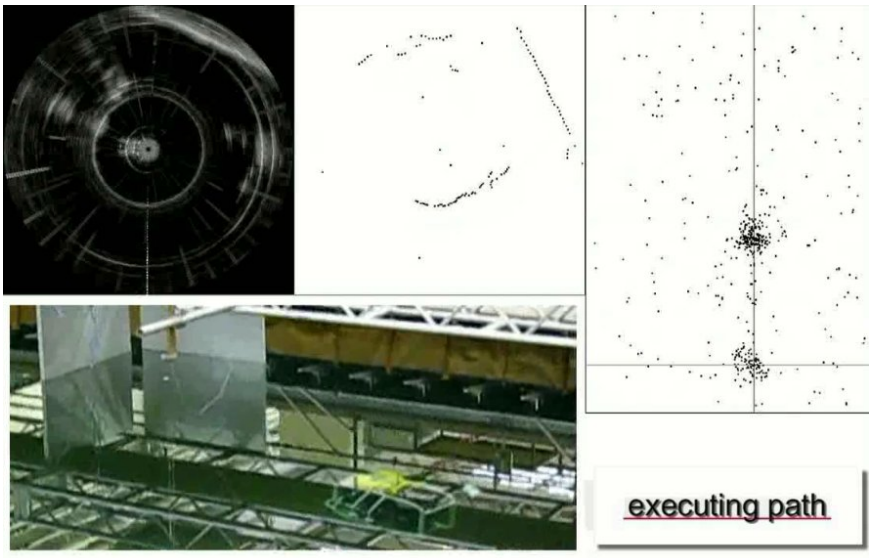


(b) Two clusters are identified. At this moment the Active Localisation module is triggered and a path is planned.

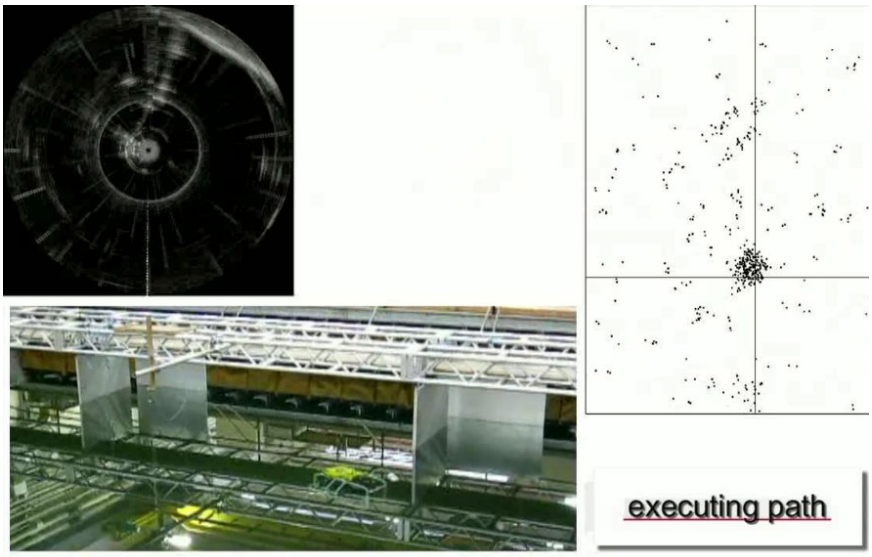


(c) Path execution I

Figure 6.12: Active localisation at the Wave Tank (1/2): raw sonar image (top left), processed sonar image (top centre), state estimation (top right), real vehicle position (centre-left).



(a) Path execution II



(b) Path execution III: particle convergence

Figure 6.13: Active localisation at the Wave Tank (2/2): raw sonar image (top left), processed sonar image (top centre), state estimation (top right), real vehicle position (centre-left).

basic actions. All the experiments presented in this Chapter are in 2D, although the full framework is perfectly usable for 3D environments. The complexity of the tree can increase going from 2D to 3D, but not necessarily. It all depends on the number of basic actions. Torpedo-shaped vehicles for example are generally underactuated, with positive surge, positive or negative pitch, and positive or negative yaw. Additionally pitch is limited and this would contribute to cut some of the tree branches. Therefore this approach is fully portable in 3D scenarios as well. For vehicles with more actuated DOFs however the complexity increases, as shown in Table 6.1. A possible way to significantly reduce the complexity is to significantly simplify the tree structure, keeping however a very similar formulation for reward/cost. Essentially, the idea is to cut branches of the tree that would bring the robot in an already visited state not only for that particular path, but also for any path already evaluated. This approach would loose the influence of previous states and the *pdf* would be only calculated related to the current node, as well as cost and reward functions. Although the data structure can be represented in the same way, to allow different config parameters to choose the level of optimisation required for the specific mission, this approach practically means to convert a tree exploration into a linear exploration, and therefore from exponential to linear complexity. Thinking about a set of actions only related to trajectory planning, this further optimisation ensures that each possible cell is visited at most once during the full execution of the algorithm (and not at most one in the same path root-node of the tree).

The proposed approach successfully addresses all the objectives set at the beginning of the Chapter. Comparison with other techniques are presented, as well as limitations and scalability issue. It therefore represents a clear contribution and advancement with respect to the state of the art.

6.5 Publications related to the Chapter

6.5.1 International Conferences & Workshops

- Y. Petillot, **F. Maurelli**; A Tree-based Planner for Active Localisation: Applications to Autonomous Underwater Vehicles; *IEEE ELMAR 2010*, Zadar, Croatia; September 2010

- **F. Maurelli**, A. Mallios, S. Krupiński, Y. Petillot, P. Ridao; Speeding-up Particle Convergence with Probabilistic Active Localisation for AUV; *IFAC Intelligent Autonomous Vehicles Symposium*, Lecce, Italy; September 2010

- **F. Maurelli**, Y. Petillot; Particle Diversity Reduction for AUV's Active Localisation; *OCEANS 2010 IEEE*; Sydney, Australia; May 2010

- **F. Maurelli**; Subakva robotiko: de homa regado al daŭra memregeco"; *Proceedings of Internacia Kongresa Universitato - IKU 2012*, Hanoi, Vietnam

Chapter 7

Conclusions and Future Work

This final Chapter aims to conclude the thesis work. It will present specific cases where this work has been applied, and will highlight and summarise the main contributions with respect to the state of the art. Finally a discussion and considerations on possible future work will end

7.1 Applications

It is very important to show applications of the presented work in real world scenarios, especially for an engineering thesis. Although results with real robots and real world conditions were already presented in the core of the thesis, this section aims to present a couple of scenarios where the work has been successfully employed.

7.1.1 Student Autonomous Underwater Challenge - Europe

The Student Autonomous Underwater Challenge Europe (SAUC-E) is a competition in which underwater robots compete in carrying out a predefined set of tasks. Held for the first time in 2006 at Pinewood Studios (UK), the event is designed to encourage students to think about underwater technology and related applications while fostering innovation and technology. Heriot-Watt University has participated in all the editions, winning the 2008 and 2009 editions. The passive localisation system described in Chapter 3 has been integrated into the vehicle *Nessie IV*, for the 2009 edition, held at the QinetiQ Ocean Basin Tank (UK). Figure 7.1 and Figure 7.2 show the results of the localisation algorithm at the Somerton Diving Pool, during SAUC-E preparation. From the Figures, it is possible to see the robustness of the algorithm, both in presence of high acoustic disturbances and limited number of particles for optimisation purposes. The algorithm was then tested in the large QinetiQ tank. The results of this testing were also successful. However the differences in the

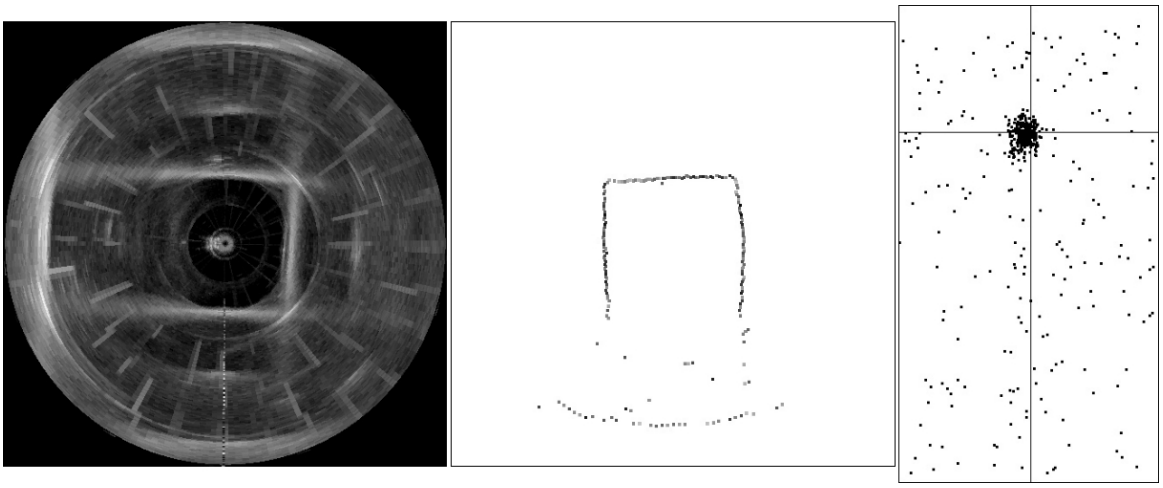


Figure 7.1: Raw sonar image, with range 10 m (left), segmented image (centre), vehicle state estimation in the environment, 6x11 m (right). It is to be noted that the raw sonar image is in sonar reference frame, with the sonar mounted with the head looking down and with a rotation of 90 deg, while the segmented image is already transformed in vehicle reference frame. The crossing lines in the right image shows the particle with greater weight. The real position of the vehicle is not known, as there is no ground-truth sensor available underwater. However, it can be inferred by looking at the raw sonar image.

environment required a reanalysis of the algorithm and its parameters. For the first time, it was tested on a very large environment, whilst all previous tests were done in relatively similar conditions. The most important factor which needed tuning was the noise level in the particle distributions, which needed to be scaled, according to the scale of the environment. Figure 7.3 shows the team at SAUC-E 2009.

7.1.2 Autonomous Inspection / Intervention Vehicles

The industrial interest for underwater inspection vehicles starts from the very beginning of the exploitation of the sea resources. At the moment most of the inspection tasks are carried out routinely by Remotely Operated Vehicles (ROVs). The interest for AUVs to carry inspection tasks can be easily understood in looking at current underwater industrial operations. An AUV allows operators to reduce the manpower by around 50% compared to current operations, where the work is carried out by ROVs. That also means that personnel logistics are reduced dramatically. Furthermore, pilots are not needed over 24 h and the on line team can be reduced dramatically, as explained by Kermorgant & Scourzic [55]. The AUV operating footprint is unlimited, since it is free swimming whereas a ROV has got limitations due to its umbilical. An AUV also enables the surface ship to carry out parallel tasks. Out of the same reason, a double speed can be achieved (0 - 6 knots with an AUV, only up to 3 knots with a

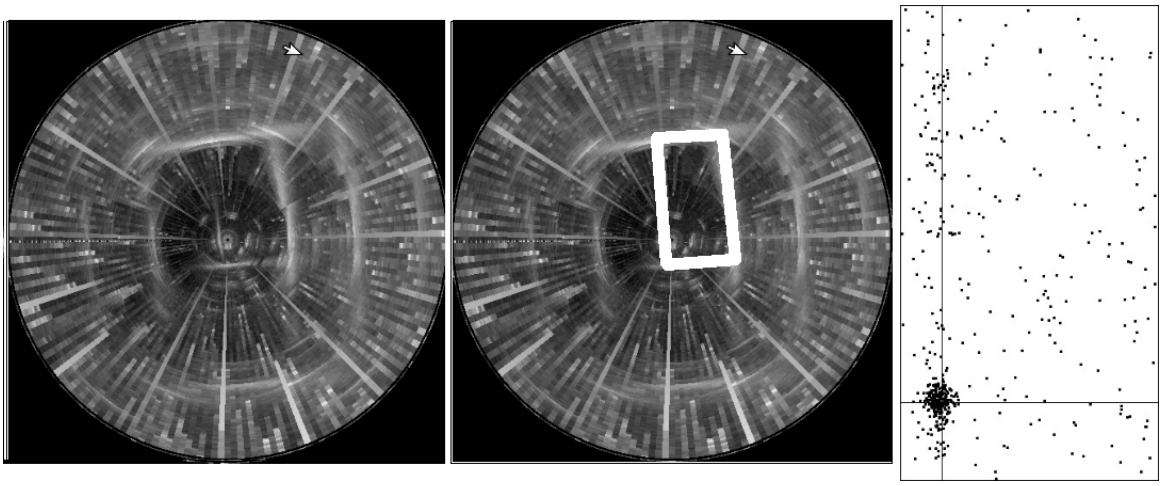


Figure 7.2: Raw sonar image, with range 20 m and with high interference noise (left), highlighted pool borders (centre), vehicle state estimation in the environment, 6x11 m (right). In addition to the noise, the image presents multipath reflections, which should not be confused with the pool borders. Despite the noise and multipath, the algorithm is very robust.

Figure 7.3: Team Nessie at SAUC-E 2009

ROV in deep water). When using an AUV the operation duration is reduced, not only because it is faster, but the vehicle not being influenced by the sea state and the ship's movements. The total weight of the AUV spread is about $1/5^{th}$ of a ROV spread. Additionally, the quality of the data retrieved by an AUV is often higher than the data retrieved by an ROV. This is due to the fact that an AUV operates without umbilical. Therefore the vehicle is neither disturbed by the ships movements transferred to the vehicle by the umbilical, nor by the umbilical own vibrations, and can stabilise itself without the constant need of a pilot, and more precisely. Autonomous localisation is essential in order to successfully perform inspection tasks. According to the type of problem, the complexity of the required localisation can change (e.g. relative to the structure to be inspected or global). Going towards persistent autonomy, active localisation is not just a nice tool, but a real need.

An Inspection/Intervention-AUV represents a new class of autonomous underwater vehicles, which are used not just for surveys, but also to inspect and interact with underwater structures. A *hybrid* ROV - or *intelligent* ROV - is generally considered an intermediate step towards I-AUVs. In the next section results on localisation performed on a I-AUV are presented.

Navigation around structures

This section presents the results of the localisation algorithm developed in Section 3.2, applied to a I-AUV. The robotic platform used for the experiments is a hover capable AUV such as the one in Figure 7.4.

The trials were performed in a cylinder tank, 8 metres tall, with a diameter of 14 metres. The first part of our validation process was to test the localisation algorithm in the empty tank, using the wall as a reference. After validating this first step, we added a cylindrical metallic object in the center of the tank in order to analyse the performances of the algorithm in the navigation around structures.

An *a-priori* map of the vehicle surroundings is known. The vehicle is equipped with a fibre optical gyroscope, DVL and depth sensor, as proprioceptive sensors. A Tritech Seaking mechanically scanned sonar is the main exteroceptive sensor which acquires range profiles of the environment.

In order to use our particle filter approach, a simulation of the sensor's view is computed as if the vehicle were located at each of the particle's position. According to the particular geometrical situation of our test facility, a geometrical approach was used, in order to simulate the sensor data, possible because both the tank and the inner object are cylinders. Other approaches, such as ray tracing, are of course possible, but more computational expensive.

In Figure 7.5, a 2D plot with the initial particle distribution is presented.

Figure 7.4: The **P**rototype **A**utonomous **I**nspection **V**ehicle - PAIV

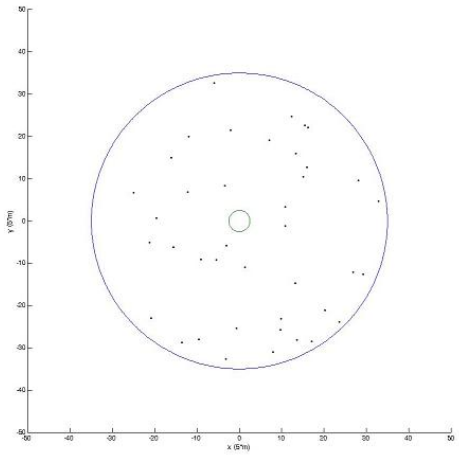


Figure 7.5: Initial distribution of particles

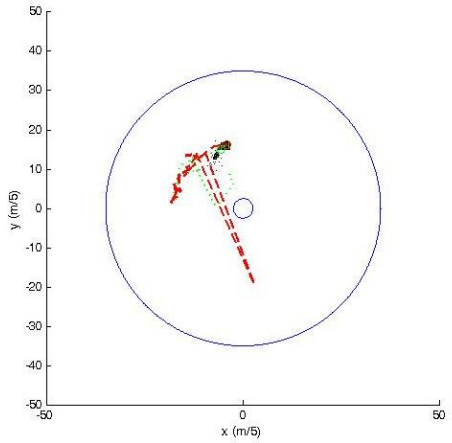


Figure 7.6: A 2D plot of the environment, with the particles, in their last configuration and with the expected trajectories, given by particle analysis. The green (light gray) dot trajectory is given by the mean of the particles, while the red (dark gray) dash one is given by the best particle.

In Figure 7.6, the expected trajectories are plotted, as well as the final configuration of the particles. The green (light gray) trajectory is given by the mean of the particles and the red (dark gray) trajectory is given by the best particle.

The general behaviour of the algorithm is similar to other tests presented in Chapter 3. There is some uncertainty at the beginning, but the particles converge to the right position quite quickly. The mission was to track the cylinder object, keeping a fixed distance from it, for about a quarter of a complete turn and then come back on the same trajectory. As shown in Figure 7.6, the proposed algorithm successfully returns the expected trajectory for the described mission.

Figure 7.7 shows the error between the real trajectory and the trajectories inferred by the particles. The green (light gray) dot error line is referred to the trajectory given by the mean of the particles, whilst the red (dark gray) dash error line is referred to the trajectory given by the best particle. The results are very good. In all our tests, the error is less than 40 cm, after the convergence. If we compare these results with the simulated data presented in 3.2, they appear to be even better. In reality, an important factor has impact on these results. The complexity of the map has to be taken into account. The synthetic environment was much more complex and bigger than the real one, presenting similar profiles, which could lead more easily to a wrong convergence.

Transit among different sites

Inspection AUV might be required to make long transit from one underwater installation to another one. When the robot travels between two different sites, the

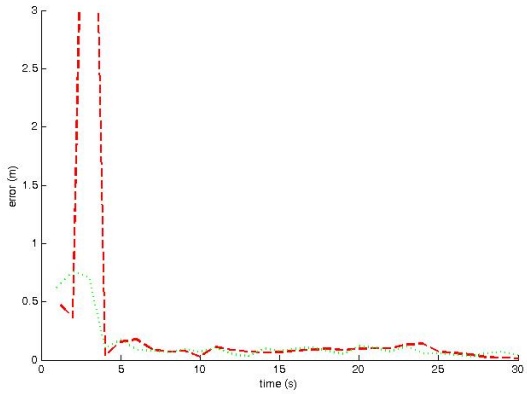


Figure 7.7: Trials with real vehicle: error between real trajectory and expected trajectories, inferred by the particles. The red (dark gray) dash error line is given by the best particle trajectory, while the green (light gray) dot error line is given by the mean trajectory.

navigation error might be bigger that the distance between two similar elements in the same site, being them pipes, raises or other infrastructures. In this context, the techniques developed in Chapter 6 would be very beneficial. After reaching the target location, the vehicle can perform active localisation to disambiguate among multiple hypotheses, and therefore can carry out its task on the right infrastructure.

7.2 Summary of the thesis

This thesis has analysed the topic of autonomous localisation for underwater vehicles.

After the Introduction, Chapter 2 has focused on *passive* techniques for AUV localisation. Several techniques have been presented, related to the state-of-the-art. The mathematical background of Bayesian filtering was outlined. Some of the techniques have been implemented and results both in simulation and with real data were presented. Chapter 3 has focused on the contribution related to passive localisation techniques. The most relevant contribution is the design and test of an improved particle filter, to be able to recover from wrong convergences. Additionally partially known maps were addressed both with particle filter and Extended Kalman Filter. Finally, an intelligent solution switching among particle filter and EKF was presented, in order to gain the advantages of both techniques.

Chapter 4 started the topic of the control system in the loop for navigation. Addressing relative navigation with respect to an underwater structure, three approaches were presented and compared.

Chapter 5 moved into a full exploration of the active localisation topic, not just

sonar servoing as in the previous chapter. Current state-of-the-art was analysed and shortfalls discussed. Chapter 6 presented a novel approach to active localisation. This system is mostly useful in case of more than one pick of probability, in the probability distribution function (*pdf*). Starting from n possible states, the planning system produces a set of actions to be executed in order to reduce the uncertainty. Experimental results have shown the reliability of this technique, both on simulated data, and in field trials. Comparison with other techniques was also successfully presented.

Finally, Conclusions are giving an overview of the thesis, highlighting applications, contributions and future work.

7.3 Summary of the main contributions

This thesis has presented advancements both for passive and active localisation systems.

Among the passive techniques, the main contributions are:

- an improved particle filter algorithm, adapted for the underwater domain, with an *ad-hoc* resampling step, able to recover from wrong convergences, showing reliability and efficiency.
- a consideration for partially known maps, with results both using Particle Filter and a modification to the standard Extended Kalman Filter.
- a novel localisation system which joins Particle Filter and Extended Kalman Filter with an intelligent switch system, according to the current state, goal and circumstances.

Considering navigation around structures, a novel and simple technique was presented and successfully evaluated compared to state-of-the-art algorithms.

Regarding the active localisation area, which probably represents the most important contribution, a novel system was presented which:

- is general enough, and not tight to any specific custom problem;
- is able to handle multiple possible locations (global localisation), and not just the current one (position tracking);
- considers the entropy expectation as an important information in the definition of the actions to be performed;

- considers multiple actions, as one action is not enough to properly address the problem;
- optimises the computational load

It was also successfully tested compared to other techniques.

Intelligent decision making, in order to solve the localisation problem, is not just an additional feature, but an essential one. Without any doubts, the research in the field of underwater robotics goes towards permanent underwater deployment, persistent autonomy, increased capabilities. Active localisation represents an important skill to reliably perform prolonged missions.

Finally, it is important to underline that all the proposed algorithms have been validated not just in simulation, but fully integrated in several robot architectures and successfully proven to be reliable in several in-water tests. No contribution that was presented in this thesis is the results of simulated numerical tests only. Everything was tested at the minimum post-processing real data, and in most of the cases fully integrated in several AUVs. This naturally resulted in an extra engineering work, which was however essential to present reliable and complete results.

7.4 Future Work

The results of this thesis have put a basis for future work in several directions. Considering the passive localisation algorithm proposed in Chapter 3, future work is related to improve the precision of the results. A way to do so is to consider the distortion in the sonar image, caused by the vehicle motion. Mechanically scanning sonar are generally slow to acquire data, and the image can be therefore very distorted, especially in cases of rotation. To cope with this problem, there are two possible ways. The first one leaves the core algorithm without changes, but changes the way the simulated image is computed. Instead of computing a 360 deg from a single position, the full motion estimate (e.g. from DVL) is considered, so that each single ping is simulated at a closer state point, than a fixed one for the full image. The second approach would require a minimal change to the algorithm itself. Instead of considering full 360 deg images, a full cycle in the particle filter algorithm could be run for each single sonar beam. This would require to change the sensor model, changing its probability function. This change is required in order to prevent the particle impoverishment problem. Considering the obvious noise in sonar images, the same precise model for laser would not work. Considering the localisation in partially known maps and the scan matching sections, future work is related to a map update through scan matching. This would allow the vehicle to both self-localise in the

environment and to update its previous knowledge with the sensor reading. It is to be noted that this problem can be considered as SLAM (Simultaneous Localisation and Mapping), although there are some important differences, which should result in an implementation less computationally expensive than the one required to solve the full SLAM problem. Future work is also related to a different field, namely solving a *semantic* localisation problem, and not just a *geometric* one. A cognitive perception of the environment is certainly useful to complete the geometric one. A localisation system based on the semantics associated to different observations can be more efficient than a mere geometric approach.

With regards to the active localisation approach, future work is related mainly to the extension in 3D. The algorithm does not require actually any structural modification. Relevant actions should be added to the set of the possible actions, like for example *dive for x meters* or *go up for x meters*. The real needed work for this extension is related to the sensor simulation, in order to evaluate the particles' weight. Several approaches can be considered, from 3D raytracing to considering several 2D maps on top of each other, "slicing" the environment. Additionally, applications in open water scenarios would be beneficial to further validate the approach. A further area of research linked to the work on active localisation is path planning, with a specific location to be reached. Whilst the work presented in this thesis was aiming at localisation only, it could be applied to look for paths arriving to a target location, with a trade-off among the cost of the path and the expectations on the features (in a general sense) perceived, in order to keep the vehicle localised. The algorithm would not start from different clusters in the *pdf*, as the original position might be known in advance, but would use similar techniques described in Chapter 6 to evaluate possible paths to the target location.

Bibliography

[1] http://www.imagenex.com/sonar_theory.pdf. Resource last accessed on 2011/12/01.

[2] Alspach, D. and Sorenson, H. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4): 439–448, Aug 1972.

[3] A. Doucet and S. Godsill and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[4] P.E. An, A.J. Healey, S.M. Smith, and S.E. Dunn. New experimental results on gps/ins navigation for ocean voyager ii auv. In *Proceedings of Symposium on Autonomous Underwater Vehicle Technology*, pages 249 –255, jun 1996.

[5] Kjetil Bergh Anonsen and Oddvar Hallingstad. Terrain aided underwater navigation using point mass and particle filters. In *Proceedings of the IEEE/ION Position Location and Navigation Symposium*, 2006.

[6] G. Antonelli, A. Caiti, V. Calabrò, and S. Chiaverini. Designing behaviors to improve observability for relative localization of auvs. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4270 –4275, may 2010.

[7] A. Arsenio and M.I. Ribeiro. Active range sensing for mobile robot localization. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1066–1071, Oct 1998.

[8] Bailey, T. and Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics &Automation Magazine*, 13(3):108–117, Sept. 2006.

[9] S. Barkby, S. Williams, O. Pizarro, and M. Jakuba. Incorporating prior maps with bathymetric distributed particle slam for improved auv navigation and mapping. In *Proceedings of MTS/IEEE OCEANS 2009 Biloxi*, pages 1–7, 2009.

[10] Rudolf Bauer. Active manoeuvres for supporting the localisation process of an autonomous mobile robot. *Robotics and Autonomous Systems*, 16(1):39 – 46, 1995. ¡ce:title¿Intelligent Robotics Systems {SIRS} '94¡/ce:title¿.

[11] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[12] S. Bhuvanagiri and K.M. Krishna. Active global localization for multiple robots by disambiguating multiple hypotheses. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3446 – 3451, sept. 2008.

[13] Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Active mobile robot localization. In *Proceedings of International Joint Conferences on Artificial Intelligence*. Morgan Kaufmann, 1997.

[14] A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo. Set-membership acoustic tracking of autonomous underwater vehicles. *Acta Acustica united with Acustica*, 5(88):648 – 652, 2002.

[15] A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo. Localization of autonomous underwater vehicles by floating acoustic buoys: a set-membership approach. *IEEE Journal of Oceanic Engineering*, 30(1):140 – 152, jan. 2005.

[16] O. Cappe, S.J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, May 2007.

[17] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings on Radar, Sonar and Navigation*, 146(1):2–7, Feb 1999.

[18] S. Carreno, P. Wilson, P. Ridao, and Y. Petillot. A survey on terrain based navigation for auvs. In *OCEANS 2010*, pages 1–7, Sept 2010.

[19] G. Chabert and L. Jaulin. QUIMPER, A Language for Quick Interval Modelling and Programming in a Bounded-Error Context. *Artificial Intelligence*, 173:1079–1100, 2009.

[20] Christopher M Clark, Christopher S Olstad, Keith Buhagiar, and Timmy Gambin. Archaeology via underwater robots: Mapping and localization within maltese cistern systems. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 662–667. IEEE, 2008.

[21] Ingemar J. Cox. Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, 1991.

[22] R. Cox and S. Wei. Advances in the state of the art for AUV inertial sensors and navigation systems. *IEEE Journal of Oceanic Engineering*, 20(4):361–366, 1995.

[23] R. Cristi. In a partially known navigation and localization environment. In *Proceedings of Symposium on Autonomous Underwater Vehicle Technology*, pages 263 –267, jul 1994.

[24] A.J. Davison and N. Kita. Active visual localisation for cooperating inspection robots. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 3, pages 1709–1715 vol.3, 2000.

[25] J. Djugash, S. Singh, and P.I. Corke. Further results with localization and mapping using range from radio. In *International Conference on Field & Service Robotics (FSR '05)*, volume 25, pages 231–242, July 2005.

[26] Gregory Dudek, Kathleen Romanik, and Sue Whitesides. Localizing a robot with minimum travel. *SIAM Journal on Computing*, 27, N. 2:583–604, 1998.

[27] Glenn R. Elion and Herbert A. Elion. *Electro-Optics Handbook*. CRC Press, 1979.

[28] M. Erol, L.F.M. Vieira, and M. Gerla. AUV-aided localization for underwater sensor networks. *Proceedings of International Conference on Wireless Algorithms, Systems and Applications*, pages 44–54, Aug. 2007.

[29] Frank Fahy. *Fundamentals of noise and vibration*. Taylor & Francis., 1998.

[30] Nathaniel Fairfield and David Wettergreen. Active localization on the ocean floor with multibeam sonar. In *Proceedings of MTS/IEEE OCEANS*, 2008.

[31] H.J.S. Feder, J.J. Leonard, and C.M. Smith. Incorporating environmental measurements in navigation. In *Proceedings of Workshop on Autonomous Underwater Vehicles*, pages 115 –122, aug 1998.

[32] B. Ferreira, A. Matos, and N. Cruz. Single beacon navigation: Localization and control of the MARES AUV. In *Proceedings of IEEE OCEANS 2010*, pages 1 – 9, sept. 2010.

[33] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207, 1998.

[34] A. Gasparri, A. Panzieri, F. Pascucci, and G. Ulivi. A hybrid active global localisation algorithm for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3148–3153, April 2007.

[35] F. Giuffrida, P. Morasso, G. Vercelli, and R. Zaccaria. Active localization techniques for mobile robots in the real world. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1312 –1318, nov 1996.

[36] F. Giuffrida, P. Morasso, G. Vercelli, and R. Zaccaria. Integration of active localization systems in vehicles control for real-time trajectory tracking. In *Proceedings of IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 549–556, Dec 1996.

[37] N. Gordon, D. Salmond, and C. Ewing. A novel approach to nonlinear nongaussian bayesian estimation. In *Proceedings of IEEE Radar and Signal Processing*, volume 40, pages 107–113, 1993.

[38] G. Grisetti, L. Iocchi, and D. Nardi. Global hough localization for mobile robots in polygonal environments. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2002.

[39] J.S. Gutmann, T. Weigel, and B Nebel. A fast, accurate, and robust method for self-localization in polygonal environments using laser-range-finders. *Advanced Robotics Journal*, 14:2001, 2000.

[40] E. Halbwachs and D. Meizel. Bounded-error estimation for mobile vehicule localization. *Proceedings of IMACS Multiconference (Symposium on Modelling, Analysis and Simulation)*, pages 1005–1010, 1996.

[41] L. Hostetler and R. Andreas. Nonlinear kalman filtering techniques for terrain-aided navigation. *IEEE Transactions on Automatic Control*, 28(3):315 – 323, mar 1983.

[42] A. Howard, M.J. Matark, and G.S. Sukhatme. Localization for mobile robot teams using maximum likelihood estimation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 434 – 439, 2002.

[43] Luyue Huang, Bo He, and Tao Zhang. An autonomous navigation algorithm for underwater vehicles based on inertial measurement units and sonar. In *2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR)*, volume 1, pages 311 –314, march 2010.

[44] L. Iocchi and D. Nardi. Hough localization for mobile robots in polygonal environments. *Robotics and Autonomous Systems*, 40:43–58, 2002.

[45] L. Jaulin. Robust set membership state estimation ; application to underwater robotics. *Automatica*, 45(1):202 – 206, 2009.

[46] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.

[47] P. Jensfelt and S. Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5):748–760, Oct 2001.

[48] Hong jian Wang, Jing Wang, Le Yu, and Zhen ye Liu. A new SLAM method based on SVM-AEKF for AUV. In *Proceedings of IEEE OCEANS 2011*, pages 1–6, 2011.

[49] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defence Sensing, Simul. and Controls, Orlando, FL*, pages 182–193, 1997.

[50] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engeneering*, 82:35–45, 1960.

[51] Kalman, R. E. and Bucy, R. S. New results in linear filtering and prediction theory. *Transactions of the ASME. Series D, Journal of Basic Engineering*, 83:95–107, 1961.

[52] G. Kantor and S. Singh. Preliminary results in range-only localization and mapping. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1818–1823 vol.2, 2002.

[53] R. Karlsson, F. Gustafsson, and T. Karlsson. Particle filtering and Cramer-Rao lower bound for underwater navigation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 65–73, April 2003.

[54] George C. Karras, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, Hashim Kemal Abdella, Tom Larkworthy, and David Lane. A robust sonar servo control scheme for wall-following using an autonomous underwater vehicle. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[55] H.A. Kermorgant and D. Scourzic. Interrelated functional topics concerning autonomy related issues in the context of autonomous inspection of underwater structures. In *Proceedings of IEEE OCEANS 2005 - Europe*, volume 2, pages 1370 – 1375, june 2005.

[56] P. Kimball and S. Rock. Sonar-based iceberg-relative AUV navigation. In *Proceedings of IEEE/OES Autonomous Underwater Vehicles*, pages 1 –6, oct. 2008.

[57] K. Kodaka, H. Niwa, and S. Sugano. Active localization of a robot on a lattice of rfid tags by using an entropy map. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3921 –3927, may 2009.

[58] H. Kondo, T. Maki, T. Ura, Y. Nose, T. Sakamaki, and M. Inaishi. Relative navigation of an AUV using image-and-acoustic based profiling systems. In *Proceedings of MTS/IEEE OCEANS*, volume 3, pages 1330 –1335, nov. 2004.

[59] H. Kondo and T. Ura. Detailed object observation by autonomous underwater vehicle with localization involving uncertainty of magnetic bearings. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 412 – 419, 2002.

[60] V. Kreinovich, L. Longpre, P. Patangay, S. Ferson, and L. Ginzburg. Outlier detection under interval uncertainty: Algorithmic solvability and computational complexity. In I. Lirkov, S. Margenov, J. Wasniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, Proceedings of the 4th International Conference LSSC'2003, 2003.

[61] Rainer Kümmerle, Patrick Pfaff, Rudolph Triebel, and Wolfram Burgard. Active monte carlo localization in outdoor terrains using multi-level surface maps.

In *Proceedings of the International Conference on Field and Service Robotics*, pages 29–35, 2007.

[62] D. Kurth, G. Kantor, and S. Singh. Experimental results in range-only localization with radio. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 974–979, Oct. 2003.

[63] H. Lahanier, E. Walter, and R. Gomeni. OMNE: a new robust membership-set estimator for the parameters of nonlinear models. *Journal of Pharmacokinetics and Biopharmaceutics*, 15:203–219, 1987.

[64] Lionel Lapierre, Rene Zapata, and Pascal Lepinay. Combined path-following and obstacle avoidance control of a wheeled robot. *Int. Journal of Robotics Research*, 26:361–375, 2007.

[65] Ji-Hong Li, Mun-Jik Lee, Sang-Hyun Park, and Jong-Geol Kim. Range sonar array based SLAM for P-SURO AUV in a partially known environment. In *Proceedings of 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 353–354, 2012.

[66] Feng Lu and E.E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 935–938, 1994.

[67] Hanjiang Luo, Yiyang Zhao, Zhongwen Guo, Siyuan Liu, Pengpeng Chen, and L.M. Ni. Udb: Using directional beacons for localization in underwater sensor networks. In *Proceedings of 14th IEEE International Conference on Parallel and Distributed Systems*, pages 551 –558, dec. 2008.

[68] S. Maeyama, A. Ohya, and S. Yuta. Positioning by tree detection sensor and dead reckoning for outdoor navigation of a mobile robot. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 653 –660, oct 1994.

[69] T. Maki, T. Matsuda, T. Sakamaki, and T. Ura. Auv navigation with a single seafloor station based on mutual orientation measurements. In *Underwater Technology (UT), 2011 IEEE Symposium on and 2011 Workshop on Scientific Use of Submarine Cables and Related Technologies (SSC)*, pages 1 –7, april 2011.

[70] Gian Luca Mariottini and Stergios I. Roumeliotis. Active vision-based robot localization and navigation in a visual memory. In *Proceedings of IEEE Inter-*

national Conference Robotics and Automation (ICRA), pages 6192 –6198, may 2011.

[71] F. Maurelli, J. Cartwright, N. Johnson, G. Bossant, P.L. Garmier, P. Regis, J. Sawas, and Y. Petillot. Nessie iv autonomous underwater vehicle. In *Proceedings of UUVS 2009, Southampton, UK*, 2009.

[72] F. Maurelli, S. Krupiński, A. Mallios, Y. Petillot, and P. Ridao. Sonar-based auv localization using an improved particle filter algorithm. In *Proceedings of IEEE OCEANS '09, Bremen, Germany*, 2009.

[73] Tiah E. McKinney. What is the biological affect of spectrophotometry of light in water? http://www.ed.mtu.edu/esmis/id65.htm. resource accessed 2008/10/07.

[74] D. Meizel, O. Lévêque, L. Jaulin, and E. Walter. Initial localization by set inversion. *IEEE Transactions on Robotics and Automation*, 18(6):966–971, 2002.

[75] D. Meizel, A. Preciado-Ruiz, and E. Halbwachs. Estimation of mobile robot localization: geometric approaches. In M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, editors, *Bounding Approaches to System Identification*, pages 463–489. Plenum Press, New York, NY, 1996.

[76] J. Minguez, L. Montesano, and F. Lamiraux. Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Transactions on Robotics*, 22(5):1047–1054, 2006.

[77] M. Montemerlo, S. Thrun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 695–701 vol.1, 2002.

[78] L. Montesano, J. Minguez, and L. Montano. Probabilistic scan matching for motion estimation in unstructured environments. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3499–3504, 2005.

[79] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA, 1979.

[80] A.C. Murtra, J.M. Mirats Tur, and Alberto Sanfeliu. Efficient active global localization for mobile robots operating in large and cooperative environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2758–2763, May 2008.

[81] P. Newman and H. Durrant-Whyte. Using sonar in terrain-aided underwater navigation. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 440 –445 vol.1, may 1998.

[82] T. Nishizawa, A. Ohya, and S. Yuta. An implementation of on-board position estimation for a mobile robot-ekf based odometry and laser reflector landmarks detection. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 395 –400 vol.1, may 1995.

[83] J.P. Norton and S.M. Verez. Outliers in bound-based state estimation and identification. *Circuits and Systems*, 1:790–793, 1993.

[84] J.M. O'Kane and S.M. LaValle. Localization with limited sensing. *IEEE Transactions on Robotics*, 23(4):704 –716, aug. 2007.

[85] C.F. Olson and L.H. Matthies. Maximum likelihood rover localization by matching range maps. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 272 –277 vol.1, may 1998.

[86] Clark F. Olson. Selecting landmarks for localization in natural terrain. *Autonomous Robots*, 12(2):201–210, 2002.

[87] E. Olson, J. Leonard, and S. Teller. Robust range-only beacon localization. In *IEEE/OES Autonomous Underwater Vehicles*, pages 66–75, June 2004.

[88] Alberto Ortiz, Javier Antich, and Gabriel Oliver. A bayesian approach for tracking undersea narrow telecommunication cables. In *Proceedings of IEEE OCEANS '09, Bremen, Germany*, 2009.

[89] Bing Ouyang, F. Dalgleish, A. Vuorenkoski, W. Britton, B. Ramos, and B. Metzger. Visualization and image enhancement for multistatic underwater laser line scan system using image-based rendering. *IEEE Journal of Oceanic Engineering*, 38(3):566–580, 2013.

[90] Y. Petillot, F. Maurelli, N. Valeyrie, A. Mallios, P. Ridao, J. Aulinas, and J. Salvi. Acoustic-based techniques for auv localisation. *Journal of Engineering for Maritime Environment*, 224(4):293–307, 2010.

[91] L. Pronzato and E. Walter. Robustness to outliers of bounded-error estimators and consequences on experiment design. In M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, editors, *Bounding Approaches to System Identification*, pages 199–212, New York, 1996. Plenum.

[92] Malvika Rao, Gregory Dudek, and Sue Whitesides. Randomized algorithms for minimum distance localization. In *Proceedings of Sixth International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 265–280, 2004.

[93] L. Reggiani and R. Morichetti. Hybrid active and passive localization for small targets. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1 – 5, sept. 2010.

[94] D. Ribas, P. Ridao, J.D. Tardós, and J. Neira. Underwater SLAM in man-made structured environments. *Journal of Field Robotics*, 25(11-12):898–921, October 2008.

[95] V. Rigaud, L. Marce, J.L. Michel, and P. Borot. Sensor fusion for auv localization. In *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology*, pages 168 –174, jun 1990.

[96] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings of the third International Conference on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

[97] A.P. Scherbatyuk. The AUV positioning using ranges from one transponder LBL. In *Proceedings of MTS/IEEE OCEANS 1995*, volume 3, pages 1620 – 1623, oct 1995.

[98] S. Seifzadeh, Dan Wu, and Yuefeng Wang. Cost-effective active localization technique for mobile robots. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 539 –543, dec. 2009.

[99] D. Silver, D. Bradley, and S. Thayer. Scan matching for flooded subterranean voids. *IEEE Conference on Robotics, Automation and Mechatronics*, 1:422–427, Dec. 2004.

[100] J. Sliwka. *Using set membership methods for robust underwater robot localization*. PhD dissertation, ENSTA-Bretagne and UBO University, 2011.

[101] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[102] James R. Solberg, Kevin M. Lynch, and Malcolm A. MacIver. Robotic electrolocation: Active underwater target localization with electric fields. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4879–4886, 2007.

[103] N. Storkensen, J. Kristensen, A. Indreeide, J. Seim, and T. Glancy. Huginâuuv for seabed survey. *Sea Technol.*, 1998.

[104] J. Swevers, C. Ganseman, D.B. Tukel, J. de Schutter, and H. Van Brussel. Optimal robot excitation and identification. *Robotics and Automation, IEEE Transactions on*, 13(5):730 –740, oct 1997.

[105] C. Tessier, C. Debain, R. Chapuis, and F. Chausse. Active perception strategy for vehicle localisation and guidance. In *Proceedings of IEEE Conference on Robotics, Automation and Mechatronics*, pages 1–6, June 2006.

[106] Donald Thomson. Acoustic positioning systems. Presented at Hydrofest 2005, Aberdeen, Scotland, 2005.

[107] S. Thrun. *Exploring Artificial Intelligence in the New Millenium*, chapter Robotic mapping: A survey. Morgan Kaufmann, 2002.

[108] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, Massachusetts London, England, 2005.

[109] T. Ura and Kangsoo Kim. On-site ins update of an auv "r2d4" by ssbl based position estimation. In *Proceedings of MTS/IEEE OCEANS 2004*, volume 3, pages 1606 – 1611, nov. 2004.

[110] T. Ura, T. Nakatani, and Y. Nose. Terrain based localization method for wreck observation auv. In *OCEANS 2006*, pages 1 –6, sept. 2006.

[111] Karstein Vestgard, Roar Hansen, Bjorn Jalving, and Odd Arild Pedersen. The hugin 3000 survey auv-design and field results. In *Proceedings of the 11th international offshore and polar engineering conference*, 2001.

[112] E.A. Wan and R. Van der Merwe. The unscented kalman filter for non linear estimation. In *Proceedings of Symposium on Adaptive Systems for Signal Processing, Commuincations and Control,*, Lake Loouise, Alberta, Canada, 2001.

[113] Chau-Chang Wang and D. Tang. Seafloor roughness measured by a laser line scanner and a conductivity probe. *IEEE Journal of Oceanic Engineering*, 34(4):459–465, 2009.

[114] Y. Watanabe, H. Ochi, and T. Shimura. The ssbl positioning for the auv with data transmission. In *Proceedings of IEEE OCEANS 2007*, pages 1 –7, 29 2007-oct. 4 2007.

[115] S.E. Webster, L.L. Whitcomb, and R.M. Eustice. Advances in decentralized single-beacon acoustic navigation for underwater vehicles: Theory and simulation. In *Proceedings of IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 1 –8, sept. 2010.

[116] G. Welch and G. Bishop. An introduction to the kalman filter (nc 27599-3175). Technical report, University of North Carolina at Chapel Hill, 2001.

[117] E. Willemenot, P.-Y. Morvan, H. Pelletier, and A. Hoof. Subsea positioning by merging inertial and acoustic technologies. In *Proceedings of IEEE OCEANS 2009 - EUROPE*, pages 1 –8, may 2009.

[118] S.B. Williams, O. Pizarro, M.V. Jakuba, I. Mahon, S.D. Ling, and C.R. Johnson. Repeated auv surveying of urchin barrens in north eastern tasmania. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 293 –299, may 2010.

[119] A.B. Willumsen, O. Hallingstad, and B. Jalving. Integration of range, bearing and doppler measurements from transponders into underwater vehicle navigation systems. In *Proceedings of IEEE OCEANS 2006*, pages 1 –6, sept. 2006.

[120] Fanlin Yang, Xiushan Lu, Yamin Dang, and Zhimin Liu. Accurate and rapid localization of an auv in an absolute reference frame using the iterative resection. In *Proceedings of IEEE OCEANS 2010 - Sydney*, pages 1 –6, may 2010.

[121] Zhiguo Yang, Wen Xu, Zhuan Xiao, and Xiang Pan. Passive localization of an autonomous underwater vehicle with periodic sonar signaling. In *Proceedings of IEEE OCEANS 2010 - Sydney*, pages 1 – 4, may 2010.

[122] Chuho Yi, Il Hong Suh, Gi Hyun Lim, and Byung-Uk Choi. Active-semantic localization with a single consumer-grade camera. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 2161 – 2166, oct. 2009.

[123] Huang Yu and Hao Yan-ling. Method of separating dipole magnetic anomaly from geomagnetic field and application in underwater vehicle localization. In *Proceedings of IEEE International Conference on Information and Automation (ICIA)*, pages 1357 –1362, june 2010.

[124] B. Zerr, G. Mailfert, A. Bertholom, and H. Ayreault. Sidescan sonar image processing for auv navigation. In *Proceedings of IEEE OCEANS 2005 - Europe*, volume 1, pages 124 – 130, june 2005.